Titre: Title:	Un mécanisme d'authentification sécurisé pour les réseaux locaux sans fil
Auteur: Author:	Mohamed Hakim Naifer
Date:	2005
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Naifer, M. H. (2005). Un mécanisme d'authentification sécurisé pour les réseaux locaux sans fil [Mémoire de maîtrise, École Polytechnique de Montréal]. PolyPublie. https://publications.polymtl.ca/7652/

UNIVERSITÉ DE MONTRÉAL

UN MÉCANISME D'AUTHENTIFICATION SÉCURISÉ POUR LES RÉSEAUX LOCAUX SANS FIL

MOHAMED HAKIM NAIFER
DÉPARTEMENT DE GÉNIE INFORMATIQUE
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES
(GÉNIE INFORMATIQUE)

AOÛT 2005

Your file Votre référence
ISBN: 978-0-494-16822-6
Our file Notre référence
ISBN: 978-0-494-16822-6

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé:

UN MÉCANISME D'AUTHENTIFICATION SÉCURISÉ POUR LES RÉSEAUX LOCAUX SANS FIL

présenté par : NAIFER Mohamed Hakim

en vue de l'obtention du diplôme de : Maîtrise ès sciences appliquées

a été dûment accepté par le jury d'examen composé de :

M. GAGNON Michel, Ph.D., président

M. FERNANDEZ José, Ph.D., membre et directeur de recherche

M. PIERRE Samuel, Ph.D., membre et codirecteur de recherche

Mme BOUCHENEB Hanifa, Doctorat., membre

Ils dirent: "Gloire à Toi! Nous n'avons de savoir que ce que Tu nous as appris. Certes c'est Toi l'Omniscient, le Sage."

Sourate 2, al-Baqara : 32

Certes, Dieu commande l'équité, la bienfaisance et l'assistance aux proches. Et Il interdit la turpitude, l'acte répréhensible et la rébellion. Il vous exhorte afin que vous vous souveniez.

Sourate 16, les Abeilles : 90

Dis: "En vérité, ma salât (prière), mes actes de dévotion, ma vie et ma mort appartiennent à Allah, Seigneur de l'Univers."

Sourate 6, al-An'am : 162

REMERCIEMENTS

J'aimerais, tout d'abord, rendre louange à Dieu le Clément, le Tout-Puissant pour sa bénédiction, sa miséricorde et son assistance, sous toutes ses formes, tout au long de ce travail, de mes études et de mon parcours dans la vie.

J'aimerais, ensuite, remercier mes directeurs de recherche MM. Samuel Pierre et José Fernandez pour leurs qualités humaines, leurs conseils judicieux et leur soutien infaillible durant les différentes étapes de ce travail.

Ma gratitude va également à l'endroit de ma famille qui, malgré la distance qui nous sépare, n'a cessé de m'encourager afin de surmonter les périodes difficiles.

Enfin, je ne saurais oublier tous mes collègues du Laboratoire de Recherche en Réseautique et Informatique Mobile (LARIM) qui, de près ou de loin, ont contribué à l'aboutissement de ce mémoire grâce à leur fraternité et à l'ambiance de travail qu'ils ont su maintenir.

RÉSUMÉ

La généralisation de la connexion à Internet et l'utilisation de plus en plus intense des systèmes mobiles et communicants offrent des possibilités nouvelles et prometteuses. Elles introduisent également un certain nombre de risques dont il faut prendre conscience et mesurer les conséquences éventuelles. Par la suite, il faut prendre les mesures adéquates pour les prévenir. La sécurité des réseaux n'est donc plus une option mais c'est devenu une obligation.

Dans nos travaux, nous nous sommes intéressés à l'une des catégories de réseaux qui a connu l'un des plus importants essors ces dernières années : les réseaux locaux sans fil. Grâce aux nombreux avantages qu'ils offrent, telles la facilité de déploiement et l'amélioration considérable de la mobilité, un grand nombre de ces réseaux, plus communément connu sous le nom de *HotSpot*, se sont même multipliés dans des endroits publics comme les hôtels, les cafés et les aéroports.

Les problèmes de sécurité dans les réseaux locaux sans fil sont nombreux, diversifiés et complexes. Prétendre les résoudre tous serait utopique. C'est dans ce contexte que nous nous sommes concentrés sur un aspect de ces problèmes, et non le moindre, qui est l'authentification.

L'authentification dans les réseaux locaux sans fil a, pendant une longue période, été basée sur WEP et très récemment sur le nouveau standard 802.11i. C'est pour cette raison que nous avons, tout d'abord, commencé par analyser conceptuellement et techniquement les modèles et protocoles présentés dans WEP et dans le nouveau standard 802.11i, ce qui nous a permis de caractériser leurs principales failles de sécurité et de mettre en évidence les nouveaux défis qu'ils soulèvent. À la lumière de ces résultats et afin de pallier ces lacunes, ce mémoire propose un mécanisme permettant d'assurer une authentification sécurisée des usagers mobiles se déplaçant dans des réseaux locaux sans fil. Ce mécanisme est constitué essentiellement de trois grandes phases : la première consiste en la découverte par un usager mobile du réseau et des

paramètres de sécurité offerts par un point d'accès. La seconde consiste en une association 802.11 entre cet usager mobile et le point d'accès considéré, avec négociation des paramètres de sécurité. La troisième et dernière consiste en une authentification *Diameter* de bout en bout. De plus, les principes et les fondements du mécanisme que nous avons spécifié et conçu constituent en soi une contribution originale. En effet, nous proposons une architecture trois tiers avec un serveur d'authentification à part entière, ce qui nous différencie des architectures actuelles de type deux tiers. Nous introduisons un concept innovateur d'authentification mutuelle entre les usagers mobiles et le serveur d'authentification. Nous proposons un concept nouveau de port contrôlé qui permet de bloquer le trafic en cas d'échec du processus d'authentification. Afin d'assurer un degré de confidentialité accru, nous utilisons une notion cryptographique de pointe, en l'occurrence les certificats numériques. De plus, notre mécanisme est le premier à utiliser un serveur *Diameter* dans le contexte des réseaux locaux sans fil. Par ailleurs, en cas d'échec du processus d'authentification notre mécanisme permet d'informer l'usager mobile de la cause de l'échec afin d'éviter qu'elle se répète de nouveau ultérieurement.

Afin de donner encore plus de valeur à notre travail, nous avons réalisé une validation formelle de notre mécanisme. Pour ce faire, nous avons, tout d'abord, modélisé à l'aide d'une machine à états finis chaque intervenant dans le processus d'authentification de notre mécanisme, c'est-à-dire l'usager mobile, le point d'accès et le serveur d'authentification en essayant de reproduire scrupuleusement le comportement de notre modèle conceptuel. Nous avons aussi modélisé un usager mobile malhonnête et un point d'accès malicieux afin de simuler le comportement de notre mécanisme en cas d'attaques. Les résultats ont été très satisfaisants. En effet, les traces d'exécution générées par le *model-cheker* UPPAAL et l'évolution globale du système dans ses différentes phases se sont révélées conformes à notre modèle conceptuel. De plus, les différents cas d'attaque étaient automatiquement détectés et évités. Enfin, à l'aide de CTL, une logique formelle, nous avons exprimé un ensemble de propriétés d'accessibilité, de vivacité, de sûreté et de non blocage représentant des contraintes

auxquelles doit répondre notre système. La vérification effective de ces propriétés avec le *model-cheker* UPPAAL nous a permis de nous assurer de l'absence de blocage dans notre mécanisme, qu'aucune situation d'incohérence entre les différents états n'est engendrée et que toutes les attaques sont bien détectées et évitées.

ABSTRACT

The generalization of Internet connection and the increasingly intense use of communicating systems and applications offer new promising possibilities. They also introduce a certain number of risks. These risks, which we should become aware, measure the possible consequences, and thereafter take the adequate actions to prevent them with full knowledge of the facts. Consequently, the safety and security of telecommunication and data-processing networks is not any more an option but it became an obligation.

In our thesis, we focused our research on Wireless Local Area Networks (WLAN) which have gained a tremendous growth and development over the last few years mainly due to there low cost, ease of deployment and user's mobility improvement. Furthermore, they are quickly becoming ubiquitous in our every day life especially in *HotSpot* areas such as airports, hotels and coffee houses.

However, wireless local area networks have also introduced a range of heightened, diversified and complex security concerns. Claiming to solve them all would be utopian. That is why we concentrated on one aspect of these concerns, and not the least, which is authentication.

Authentication in WLAN was, for a long period, based on WEP and very recently on the new standard 802.11i. For this reason we, first of all, started by analyzing conceptually and technically the models and protocols presented in WEP and in the new standard 802.11i, which enabled us to characterize their main concerns and to put forward the new challenges they raise. In order to mitigate these problems, this thesis proposes a mechanism to ensure a secure authentication of mobile users in WLAN. This mechanism is basically divided in three main steps: the first one is to discover, by a mobile user, the network and the security parameters offered by an access point. The second one is an 802.11 association between the mobile user and the access point considered involving security parameters negotiation. The third and last one is a

Diameter authentication. The mechanism we specified and designed is our original contributions. We propose a three part architecture with an independent authentication server, which differentiates us from current two part architectures. We introduce an innovative concept of mutual authentication between the mobile users and the authentication server. We propose a new concept of controlled port which can block the traffic in case the authentication process fails. In order to ensure an increased degree of confidentiality we use a main cryptographic concept which is digital certificate. Beside that, our mechanism is the first to use a Diameter authentication server in the context of WLAN. Finally, in case of authentication failure our mechanism informs the mobile user of the reason of the failure in order to prevent it later on.

In order to give more value to our work, we carried out a formal validation of our mechanism. We first modelled each element of the authentication process, i.e. the mobile user, the access point and the authentication server using a finite states machine while trying carefully to reproduce the behaviour of our conceptual model. We also modelled a hacker mobile user and a malicious access point in order to simulate the behaviour of our mechanism in case of attacks. The results were very interesting. The execution traces generated by the UPPAAL model-checker and the global evolution of the system in its various phases were conform to our conceptual model. Moreover, the various cases of attack were automatically detected and avoided. Lastly, using CTL a formal logic we expressed some reachability, liveness, safety and non-blocking properties representing constraints of our system. The effective checking of these properties with UPPAAL allowed us to be convinced of the absence of blocking errors and inconsistency situation in our mechanism. Further more, all the attacks were well detected and avoided.

TABLE DES MATIÈRES

LISTE DES FIGURES

LISTE DES TABLEAUX

CHAPITRE I

INTRODUCTION

Durant ces dernières années, le développement des nouvelles technologies de l'information et des communications associé à la complexité croissante des échanges d'informations inter et intra entreprises a engendré un engouement certain pour l'informatisation et le développement de réseaux informatiques. De tels réseaux devraient être capables de gérer tous les flux de données qui transitent à travers l'entreprise, tout en assurant un degré de sécurité et de confidentialité essentiel au bon fonctionnement de cette entreprise. Les réseaux locaux sans fil ou Wireless Local Area Network (WLAN) sont l'une des catégories de réseaux qui ont connu un très grand essor de par les nombreux avantages qu'ils offrent. Bon nombre de ces réseaux se sont même multipliés dans des endroits publics comme les cafés, les hôtels et les aéroports, plus communément connus sous le nom de *HotSpot*. Les WLAN ont résolu beaucoup de problèmes comparativement aux réseaux locaux filaires classiques. Cependant, ils en ont introduit d'autres aussi, et notamment ceux relatifs à la sécurité des communications, qui représentent le thème du présent mémoire. Dans ce chapitre, nous allons commencer par un bref aperçu des concepts de base des réseaux sans fil et plus spécifiquement des différentes technologies utilisées pour les WLAN, puis nous définirons les éléments de la problématique suivis des objectifs de recherche. Enfin, nous terminerons par le plan de notre mémoire.

1.1 Définitions et concepts de base

Un réseau sans fil (wireless network) est un réseau dans lequel au moins deux terminaux peuvent communiquer sans liaison filaire. Grâce aux réseaux sans fil, un utilisateur a la possibilité de rester connecté tout en se déplaçant dans un périmètre géographique plus ou moins étendu. Les réseaux sans fil sont basés sur des liaisons utilisant des ondes électromagnétiques en lieu et place des câbles habituels. Il existe

plusieurs technologies se distinguant d'une part par la fréquence d'émission utilisée et d'autre part par le débit et la portée des transmissions.

Les réseaux sans fil permettent de relier très facilement des équipements distants d'une dizaine de mètres à quelques kilomètres. De plus, l'installation de tels réseaux ne demande pas de lourds aménagements des infrastructures existantes, comme c'est le cas avec les réseaux filaires, ce qui a valu un développement rapide de ce type de technologies.

En contrepartie se pose le problème de la réglementation relative aux transmissions radio. En effet, les transmissions radio servent à un grand nombre d'applications (militaires, scientifiques, amateurs, etc.) et sont sensibles aux interférences. C'est la raison pour laquelle une réglementation est nécessaire dans chaque pays afin de définir les plages de fréquences et les puissances auxquelles il est possible d'émettre pour chaque catégorie d'utilisation.

On distingue habituellement deux catégories de réseaux sans fil, selon le périmètre géographique considéré : *les réseaux personnels sans fil* et *les réseaux locaux sans fil.*

Les réseaux personnels sans fil noté **WPAN** pour *Wireless Personal Area Network* sont des réseaux sans fil d'une faible portée, de l'ordre de quelques dizaines de mètres. Ce type de réseau sert généralement à relier des périphériques (imprimante, téléphone portable, appareils domestiques, etc.) ou un assistant personnel (PDA) à un ordinateur sans liaison filaire, ou bien à permettre la liaison sans fil entre deux machines très peu distantes. Il existe plusieurs technologies utilisées pour les WPANs, la principale demeure la technologie **Bluetooth**, proposant un débit théorique de 1 Mbps pour une portée maximale d'une trentaine de mètres. Bluetooth, connue aussi sous le nom de IEEE 802.15.1, possède l'avantage d'être très peu gourmande en énergie, ce qui la rend particulièrement adaptée à une utilisation au sein de petits périphériques. Les liaisons infrarouges sont aussi une technologie WPAN permettant de créer des liaisons sans fil de quelques mètres avec des débits pouvant monter à quelques mégabits par

seconde (Mbps). Cette technologie est largement utilisée pour la domotique mais souffre toutefois des perturbations dues aux interférences lumineuses.

Les réseaux locaux sans fil noté **WLAN** pour *Wireless Local Area Network* sont des réseaux permettant de couvrir l'équivalent d'un réseau local d'entreprise, soit une portée d'environ une centaine de mètres. Ils permettent de relier entre eux les terminaux présents dans la zone de couverture. Il existe deux technologies concurrentes : **HiperLAN2** et **Wi-Fi.**

HiperLAN2 (*HIgh Performance Radio LAN 2.0*), norme européenne élaborée par l'ETSI (European Telecommunications Standards Institute), permet d'obtenir un débit théorique de 54 Mbps sur une zone d'une centaine de mètres dans la gamme de fréquences comprise entre 5 150 et 5 300 MHz.

Wi-Fi (Wireless Fidelity) est une technologie basée sur le standard 802.11 de l'IEEE (Institute of Electrical and Electronics Engineers) dont le but est de créer une norme pour les communications sans fil, permettant le développement et l'expansion des WLANs. Vu l'intérêt croissant pour ce type de technologies, plusieurs sous-familles ont été créées et se différencient par la fréquence utilisée, le débit ou la technologie de transmission :

- **IEEE 802.11a** utilise la modulation OFDM (Orthogonal Frequency Division Multiplexing) pour la transmission sur la bande de fréquences UNII (Unlicensed National Information Infrastructure) de 5.150 à 5.725 GHz et offre un débit maximal de 54 Mbps.
- **IEEE 802.11b** utilise la modulation DSSS (Direct Sequence Spread Spectrum) pour la transmission sur la bande de fréquences ISM (Industrial, Scientific and Medical) de 2.4 à 2.5 GHz, possède 3 canaux non interférant et offre un débit maximal de 11 Mbps. Par contre, les réseaux 802.11b ne sont pas compatibles avec les réseaux 802.11a.

- **IEEE 802.11g** combine les avantages de 802.11a et 802.11b. Il offre un débit maximal de 54 Mbps et utilise la modulation OFDM (Orthogonal Frequency Division Multiplexing) pour la transmission mais sur la bande de fréquence ISM (Industrial, Scientific and Medical) de 2.4 à 2.5 GHz. De plus, il est compatible avec 802.11b, ce qui permet d'augmenter le débit tout en évitant une mise à jour matérielle complète, trop coûteuse, des réseaux WLAN existants.
- **IEEE 802.11e** est un groupe de travail sur l'amélioration de la qualité de service dans les réseaux locaux sans fil (le standard n'a pas encore été approuvé), qui attribue une classe de priorité pour chaque type de trafic en utilisant la technique *Enhanced Distributed Coordination Function* (EDCF).

1.2 Éléments de la problématique

De par leur nature même, les ondes hertziennes sont difficiles à confiner dans une surface géographique restreinte, il est donc facile pour une personne malveillante d'intercepter le trafic du réseau si les informations circulent en clair. Par conséquent, il est fondamental de mettre en place des dispositifs adéquats de manière à assurer la confidentialité et l'intégrité des données circulant sur les réseaux sans fil.

La sécurité des communications dans les réseaux locaux sans fil a, pendant une longue période, été basée sur un mécanisme appelé **WEP** pour *Wired Equivalent Privacy*. Par la suite, plusieurs études [2] [3] [4] [7] [8] ont révélé des failles dans WEP, ce qui a contraint la communauté scientifique et industrielle à proposer plusieurs solutions intermédiaires, relativement efficaces, afin de remédier rapidement à ces failles, vu l'intérêt croissant et le développement exponentiel de l'utilisation des réseaux locaux sans fil. Ces failles peuvent être répertoriées en deux grandes catégories : les failles cryptographiques et les failles architecturales.

Les failles cryptographiques sont relatives à une mauvaise utilisation dans WEP de l'algorithme de cryptage **RC4**, à une mauvaise gestion des clés de cryptage et à une trop faible authentification des usagers mobiles. Les failles architecturales correspondent

à une mauvaise conception du WLAN, notamment le choix de l'emplacement des points d'accès, de l'étendue de leur signal et de leur configuration intrinsèque.

Récemment, l'IEEE a ratifié un nouveau standard **802.11i** [14] destiné à améliorer la sécurité des communications dans les réseaux locaux sans fil. Or, ce standard est basé sur plusieurs mécanismes, protocoles et architectures de sécurité qui ont fait l'objet de certaines critiques quant à leur efficacité dans le contexte des réseaux locaux sans fil. En particulier, l'architecture recommandée par l'IEEE, afin d'assurer un meilleur contrôle d'accès et une meilleure authentification, demeure vulnérable à des attaques de type usurpation d'identité ou de type mascarade [15]. L'existence de ce type de faille prouve que des améliorations peuvent être apportées au processus d'authentification utilisé dans les réseaux locaux sans fil.

1.3 Objectifs de recherche

L'objectif principal de notre recherche est de proposer un mécanisme permettant d'assurer convenablement la sécurité des communications et plus précisément l'authentification des usagers dans les réseaux locaux sans fil. De manière plus spécifique, cette recherche vise à :

- Analyser conceptuellement et techniquement les modèles et protocoles présentés dans WEP et dans le nouveau standard 802.11i afin de caractériser leurs principales failles de sécurité ;
- Spécifier et concevoir un mécanisme permettant une authentification adéquate des usagers mobiles se déplaçant dans des réseaux locaux sans fil ;
- Valider formellement notre mécanisme afin de vérifier sa robustesse eu égard aux différentes attaques et de déceler d'éventuelles erreurs de blocage ou de divergence.

1.4 Plan du mémoire

Suite à ce chapitre d'introduction, nous consacrerons le deuxième chapitre à une présentation non exhaustive des différentes failles de sécurité relatives aux réseaux locaux sans fil recensés dans la littérature, puis nous analyserons les solutions logicielles, matérielles et architecturales proposées pour y remédier. Le troisième chapitre sera consacré à la spécification et à la conception de notre mécanisme d'authentification des usagers mobiles, mécanisme basé sur les modèles et protocoles recommandés par l'IEEE et l'IETF. Ensuite, le chapitre quatre sera consacré à la validation formelle de ce mécanisme et à l'analyse des résultats qui en découlent. Enfin, le chapitre cinq résumera les principales contributions de ce mémoire, fera état des limitations de notre mécanisme et des extensions possibles aux travaux entrepris.

CHAPITRE II

ANALYSE DES MÉCANISMES DE SÉCURITÉ DANS LES WLAN

Les réseaux locaux sans fil WLAN (Wireless Local Area Network) sont l'une des catégories de réseaux qui, ces dernières années, ont connu un très grand essor de par les nombreux avantages qu'ils offrent. Ce grand succès a été matérialisé par une multiplication abondante de ce genre de réseau dans des endroits publics comme les cafés, les hôtels et les aéroports, réseau plus communément connu sous le nom de *HotSpot*. Les WLAN ont résolu beaucoup de problèmes comparativement aux réseaux locaux filaires classiques mais ils en ont introduits d'autres aussi et notamment ceux relatifs à la sécurité. De nombreux travaux de recherche ont étudié l'aspect sécuritaire des réseaux locaux sans fil mais aucun d'entre eux n'a réussi à résoudre totalement tous les problèmes. Dans ce chapitre, nous allons en faire une synthèse. Après avoir défini les concepts de base de la sécurité des réseaux, nous présentons une revue sélective des travaux recensés dans la littérature qui traitent d'un ou de plusieurs aspects de la sécurité des réseaux locaux sans fil.

2.1 Concepts de base de la sécurité des réseaux

L'informatique et les réseaux de télécommunications sont devenus des outils de travail indispensables pour les tâches critiques de la vie professionnelle. Ce sont des outils vitaux pour le bon fonctionnement de toute entreprise qui doit en assurer l'évolution, la progression et la sécurité, si elle veut garantir son développement. Bon fonctionnement signifie comme conditions :

- la protection des systèmes et des données nominatives ;
- la fiabilité des logiciels et des matériels ;

- la performance et la disponibilité du service (rien de plus gênant et problématique que de ne pas pouvoir utiliser un service quand on en a besoin) ;
- la bonne protection des informations stockées et échangées (intégrité d'une part, confidentialité d'autre part) ;
- la bonne protection des accès aux systèmes (seules les personnes autorisées peuvent y accéder) ;
- la réelle confiance dans l'identité des correspondants avec lesquels on échange des informations (garantie d'authentification d'une part, et assurance de non usurpation d'identité d'autre part).

Jusqu'au début des années 80, la centralisation des moyens informatiques (quelques gros serveurs par entreprise) et la quasi-absence de communication avec l'extérieur, permettaient facilement de garantir les objectifs ci-dessus grâce à une administration centralisée bien identifiée. Les temps ont changé et une entreprise ne peut plus s'isoler si elle veut profiter du déploiement de l'Internet, du commerce électronique ou des services associés aux réseaux étendus. Par contre, ces portes d'entrée sur le réseau mondial sont autant de risques d'attaques par un pirate ou de risques de mauvaises manipulations, et le bon fonctionnement de cette informatique distribuée peut être facilement perturbé [5]. En effet, si la généralisation de la connexion à Internet des entreprises et l'utilisation de plus en plus intense des systèmes et applications communicants offrent des possibilités nouvelles et prometteuses, elles introduisent également un certain nombre de risques dont il faut prendre conscience, en mesurer les conséquences éventuelles, et en connaissance de cause prendre les mesures adéquates pour les prévenir.

La sécurité informatique consiste donc, d'une manière générale, à assurer que les ressources matérielles ou logicielles d'une organisation sont uniquement utilisées dans le cadre où il est prévu qu'elles le soient [6]. La sécurité informatique couvre généralement trois principaux objectifs :

- l'intégrité, c'est-à-dire garantir que les données sont bien celles qu'on croit être ;
- la confidentialité, consistant à assurer que seules les personnes autorisées ont accès aux ressources ;
- la disponibilité, permettant d'accéder aux données et aux services à tout moment et par suite de maintenir le bon fonctionnement du système.

La sécurité informatique doit toutefois être étudiée de telle manière à ne pas empêcher les utilisateurs de développer les usages qui leur sont nécessaires, et de faire en sorte qu'ils puissent utiliser les systèmes en toute confiance. C'est la raison pour laquelle il est nécessaire de définir dans un premier temps une politique de sécurité, c'est-à-dire :

- élaborer des règles et des procédures à mettre en œuvre dans les différents services d'une entreprise ;
- définir les actions à entreprendre et les personnes à contacter en cas de détection d'une intrusion ;
- sensibiliser les utilisateurs aux problèmes liés à la sécurité des systèmes d'informations.

La politique de sécurité est donc l'ensemble des orientations suivies par une entreprise en terme de sécurité. À ce titre, elle se doit d'être élaborée au niveau de la direction de l'organisation concernée, car elle intéresse tous les utilisateurs du réseau [6]. La sécurité des réseaux informatiques fait souvent l'objet de métaphores. Ainsi, on la compare régulièrement à une chaîne en expliquant que le niveau de sécurité d'un système est caractérisé par le niveau de sécurité du maillon le plus faible. Par suite, une porte blindée est inutile dans un bâtiment si les fenêtres sont ouvertes sur la rue. Cela signifie que la sécurité doit être abordée dans un contexte global.

2.2 Défis de sécurité dans les WLAN

Les défis de sécurité dans les WLAN ont fait l'objet de plusieurs recherches et ont donnée lieu à diverses publications [2] [3] [4] [7] [8]. Dans la suite, nous présentons, tout d'abord, les vulnérabilités découvertes au sein du mécanisme cryptographique utilisé dans les WLAN, en l'occurrence WEP (Wired Equivalent Privacy). Ces failles ont été utilisées comme base pour réaliser différentes attaques sur les réseaux locaux sans fil, nous les présenterons aussi par la suite. Enfin, nous parlerons des problèmes que peuvent engendrer les configurations par défaut des points d'accès, l'étendue de leur signal et la planification de leur emplacement.

2.2.1 Analyse de WEP et de ses vulnérabilités

Le WEP a été défini dans le standard IEEE 802.11 [1]. Son objectif principal est, comme son nom l'indique, de sécuriser les réseaux locaux sans fil au même degré d'efficacité que les réseaux câblés. Comme l'illustre la Figure 2.1 [1], WEP est basé sur l'algorithme de chiffrement symétrique RC4 (Rivest's Cipher 4) qui est un algorithme utilisant le cryptage du flux de données (stream cipher algorithm), en ayant recours à un générateur de nombres pseudo aléatoires PRNG (Pseudo Random Number Generator).

WEP requière une clé secrète devant être déployée aux différents points d'accès et dans les appareils sans fil des usagers mobiles (Laptop, PDA,...). Cette clé sera utilisée d'une part pour le cryptage des données avant leur transmission et d'autre part pour la vérification de l'intégrité des données. Généralement, elle est de 40 bits, ce qui est relativement petit par rapport aux autres protocoles de cryptage, quoiqu'une version de WEP développée par certains constructeurs permet des clés de 104 bits. Par contre, le standard ne précise pas de méthodes pour le déploiement de la clé secrète [3]. En pratique, la majorité des installations déploient une même clé pour tous les usagers mobiles et les points d'accès.

WEP utilise aussi un vecteur d'initialisation (VI) de 24 bits, qui est concaténé à la clé secrète afin de former le *seed* du PRNG (le VI représente les bits de poids faible), ce qui nous permet d'obtenir la clé RC4. D'autre part, les messages à envoyer sont utilisés pour obtenir un ICV (Integrity Check Value) grâce à l'algorithme CRC-32 (Cyclic Redundancy Code 32) afin de pouvoir vérifier l'intégrité des données transmises. Finalement, le message crypté est obtenu suite à une opération de XOR entre la clé RC4 générée par le PRNG et le message à envoyer concaténé à l'ICV.

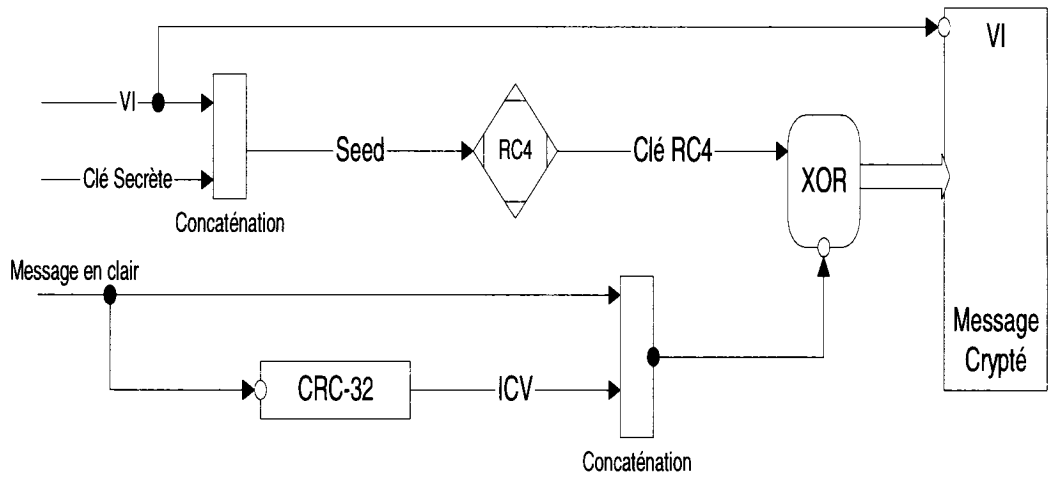

Figure 2.1 Diagramme d'encryption avec WEP

Par ailleurs, le vecteur d'initialisation est envoyé en clair au receveur afin qu'il puisse générer la même clé RC4. Par conséquent, les 24 premiers bits de tous les messages envoyés selon le protocole WEP peuvent être connu par toute personne bienveillante ou malveillante qui intercepte les messages.

D'un point de vue mathématique, ce processus peux être modélisé comme suit [4]: si nous notons M le message à envoyer et $c(M)$ l'ICV obtenu à partir de M, alors le message effectivement crypté sera :

$P = <M, c(M)>$, la concaténation des deux.

D'autre part, la clé RC4 qui dépend de k, la clé secrète, et du vecteur d'initialisation noté v sera RC4(v,k). Donc, notre message crypté *C* obtenu suite à l'opération de XOR sera :

$$C = P \oplus RC4(v,k)$$

Finalement, la transmission de l'émetteur A vers le receveur B sera symbolisé comme suit :

$$A \rightarrow B: v, (C) = v, (P \oplus RC4\ (v,k))$$

avec $P = <M, c(M)>$ et v le vecteur d'initialisation en clair.

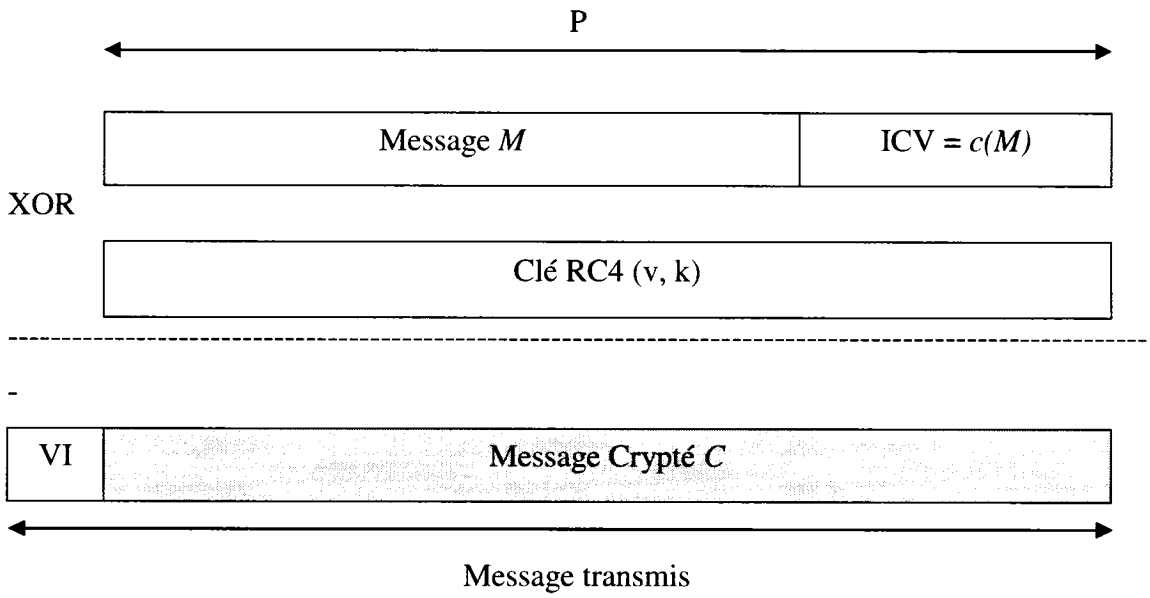

Figure 2.2 Encryption d'un message avec WEP

Le processus de décryption, illustré à la Figure 2.3, fait intervenir une concaténation du VI et de la clé secrète comme *seed* du PRNG afin de générer la bonne clé RC4, qui permettra suite à une opération de XOR avec le message crypté d'obtenir le message original et l'ICV [1]. Par suite, la phase de vérification de l'intégrité des données est enclenchée. Une nouvelle valeur ICV' est calculée à partir du message déchiffré et elle est comparée à la valeur de l'ICV reçu. Un cas de non concordance prouverait que les données originales ont été altérées.

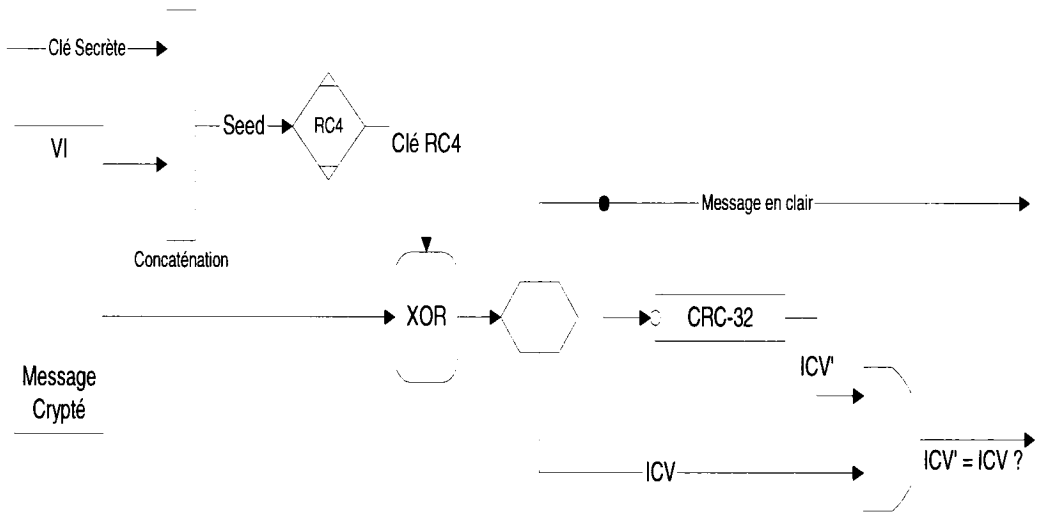

Figure 2.3 Diagramme de décryption avec WEP

Si nous suivons les même notations mathématiques précédentes, le processus de décryption sera comme suit. Sachant que le receveur B a reçu le VI en clair et le message crypté C, il devra d'abord générer la clé RC4(v,k) et ensuite on aura :

$$P' = C \oplus RC4(v,k)$$

$$= (P \oplus RC4(v,k)) \oplus RC4(v.k)$$

$$= P = \langle M', c' \rangle$$

P' est alors divisé sous la forme $\langle M', c' \rangle$ afin de calculer un ICV' noté *c(M')* qui sera comparé au *c'* reçu, ce qui permettra à B de vérifier l'intégrité des données reçues.

Le fait que le vecteur d'initialisation ne soit que de 24 bits représente un grand problème car il est pratiquement garanti que le même VI sera réutilisé pour plusieurs messages, sachant qu'un nouveau VI est utilisé pour chaque paquet. En fait, un VI pourrait être réutilisé en aussi peu de temps qu'une demi-journée s'il y a assez de trafic dans le réseau local sans fil. Certains constructeurs vont même à utiliser un unique VI pour toutes les clés générées ce qui facilite encore plus le travail d'un espion [2].

D'autre part, la gestion des clés avec WEP pose un autre type de problème : le fait qu'une clé peut être utilisée pendant une longue période sans être changée peut nuire considérablement, notamment dans le cas où un appareil sans fil est perdu ou volé, la clé secrète pourrait être récupérée et utilisée pour compromettre le réseau [2].

2.2.2 Description d'attaques sur les WLAN

WEP a été conçu afin d'assurer trois objectifs principaux de la sécurité informatique : la confidentialité, l'intégrité des données et le controle d'accès. L'atteinte de ces objectifs a été basée sur la difficulté de découvrir la clé secrète avec une attaque de type force brute.

En réalité, il y a eux deux classes d'implémentation de WEP : le WEP classique comme présenté dans le standard IEEE 802.11 [1] et une version étendue développée par quelques constructeurs. Le standard spécifie l'utilisation de clés de 40 bits. Or cette taille est trop petite pour résister à une attaque de force brute, même avec des ressources informatiques modestes [4]. Par la suite, certains constructeurs ont étendu le protocole afin d'utiliser des clés plus larges et on parle alors de WEP-128, qui en réalité n'utilise qu'une clé de 104 bits à cause du VI de 24 bits. WEP-128 rend l'attaque de type force brute plus difficile à réaliser. Par contre, certaines études montrent qu'il reste toujours vulnérable à d'autres types d'attaques que nous présentons à la suite.

1. Attaque basée sur la réutilisation de clés

Cette attaque est basée sur le fait que deux messages cryptés avec la même clé RC4 peut révéler des informations sur les deux messages. En effet, si nous avons :

$$C_1 = P_1 \oplus RC4(v,k)$$

$$\text{Et } C_2 = P_2 \oplus RC4(v,k)$$

$$\text{Alors } C_1 \oplus C_2 = (P_1 \oplus RC4(v,k)) \oplus (P_2 \oplus RC4(v,k))$$

$$= P_1 \oplus P_2$$

En d'autres termes, une opération de XOR entre deux messages cryptés C_1 et C_2 permet d'obtenir le XOR des deux messages en clair ($P1 \oplus P_2$).

À ce stade, deux cas se présentent :

1) L'un des messages en clair (P1 ou P_2) est connu. Ce cas peut être prémédité par des *hackers* en ayant recours à diverses techniques comme par exemple en envoyant du trafic IP à partir d'une station sous leur contrôle quelque part sur le réseau Internet et à destination d'un poste précis du réseau sans fil. Ou encore en envoyant des emails de *Spam* sachant que le destinataire consulte sa boite aux lettres électronique à travers un lien sans fil. Ce genre de situation peut même arriver inconsciemment dans le cas où le réseau sans fil contient des clients mobiles mixtes, supportant et ne supportant pas le cryptage WEP. Car, dans ce cas, les paquets *broadcast* qui doivent être acheminés à tout le monde seront automatiquement en clair. Toutes ces situations permettraient facilement à un *hacker* de décrypter tous les messages utilisant le même vecteur d'initialisation, à partir du moment où il intercepte la version cryptée d'un message qu'il connaît [3] [4].

2) Dans le cas plus général où l'on ne connaît ni P_1 ni P_2, diverses techniques cryptographiques connues basées sur la redondance d'information, comme l'analyse fréquentielle par exemple, permettent d'obtenir P_1 et P_2 [4].

Comme nous le constatons, ce type d'attaque est basé sur la réutilisation de la même clé RC4 pour le cryptage, qui est dû à une mauvaise gestion des vecteurs d'initialisation. En effet, vu que la clé secrète k est très rarement modifiée, une réutilisation du VI provoque, dans la majorité des cas, une réutilisation de la clé RC4 [4]. De plus, les VI sont transmis en clair, donc une duplication de VI peut facilement être détectée par un *hacker* et par suite expose directement le réseau à ce genre d'attaque.

2. Attaque de type dictionnaire

Cette attaque consiste à créer une table de correspondance entre chaque vecteur d'initialisation possible et la clé RC4 qui lui correspond. Le VI étant de 24 bits, le nombre maximum de possibilités sera de 2^{24}, ce qui requière une capacité de stockage de l'ordre de 24 GB. Cela dit, certaines cartes sans fil PCMCIA font une remise à zéro du VI à chaque fois qu'elles sont réinitialisées (c'est-à-dire au moins une fois par jour) et l'incrémentent de 1 pour chaque nouveau paquet transmis. De plus, ce type de cartes se réinitialise à chaque fois qu'elles sont insérées dans le laptop. Par conséquent, les clés RC4 correspondant à de petites valeurs de VI vont être réutilisées plus fréquemment, ce qui rend la taille de la table encore plus petite [4].

Après un certain moment, un *hacker* constituera un dictionnaire de décryption lui permettant de déchiffrer immédiatement tout message intercepté. De plus, cette attaque reste valable indépendamment d'une augmentation de la taille de la clé secrète de cryptage k, car elle dépend de la taille du vecteur d'initialisation fixé par le standard à 24 bits.

3. Attaque avec modification de message

Cette attaque permet de modifier les messages en transit sans être détectée. Elle est basée sur la propriété de linéarité des algorithmes de *checksum* (séquence de contrôle), c'est-à-dire que l'opération de *checksum* est distributive par rapport à l'opération de XOR :

$$c(x \oplus y) = c(x) \oplus c(y), \ \forall x \text{ et } \forall \ y$$

La conséquence de cette propriété est qu'il devient possible de faire des modifications contrôlées à un message crypté sans modifier la valeur du *checksum* comme le prouve le raisonnement suivant [4].

Notons C un message crypté venant de A et intercepté avant qu'il n'atteigne son destinataire B. C correspond à un message M inconnu :

$$A \rightarrow B: <v, C>$$

$$C = RC4\ (v, k) \oplus <M, c\ (M)> \quad (1)$$

Il est possible de trouver un nouveau message crypté C' qui sera décrypté en M'.

$M' = M \oplus \Delta$, Δ étant choisi arbitrairement par le *hacker*.

Après modification, le message transmis sera C' :

$$A \rightarrow B: <v, C'>$$

Lors de la décryption, le destinataire B obtiendra le message modifié M' avec le bon *checksum*. Il reste à montrer comment obtenir C' à partir de C.

Suite à une opération de XOR du terme $<\Delta, c\ (\Delta)>$ avec les deux membres de l'équation (1), nous obtenons :

$$C' = C \oplus <\Delta, c(\Delta)>$$

$$= RC4(v,k) \oplus <M,c(M)> \oplus <\Delta,c(\Delta)>$$

$$= RC4(v,k) \oplus <M \oplus \Delta, c(M) \oplus c(\Delta)>$$

$$= RC4(v,k) \oplus <M', c(M \oplus \Delta)>$$

$$= RC4(v,k) \oplus <M', c(M')>$$

À l'étape 4, nous avons utilisé la propriété de linéarité du *checksum* :

$$c(M) \oplus c(\Delta) = c(M \oplus \Delta)$$

Par la suite, nous avons montré qu'une modification aléatoire d'un message crypté C est réalisable sans pour autant connaître le message en clair M, il suffit juste de savoir le message crypté original C et la modification souhaitée Δ afin de calculer :

$$C' = C \oplus <\Delta, c(\Delta)>$$

Par exemple, pour modifier le premier bit d'un message, un *hacker* peut mettre :

$$\Delta = 1000....0$$

En conclusion, cette attaque montre que WEP ne répond pas à la propriété d'assurance de l'intégrité des données.

4. Attaque avec injection de message

Cette attaque permet à un *hacker* d'injecter des messages dans le réseau ; elle est basée sur le fait que la connaissance d'un texte en clair et de son équivalent crypté permet de déterminer la clé RC4 qui pourra être réutilisée pour crypter un nouveau paquet avec le même vecteur d'initialisation [4].

Si on note P = <M,c(M)> le message en clair et C son équivalent crypté, alors :

$$P \oplus C = P \oplus (P \oplus RC4(v,k)) = RC4(v,k)$$

Le *hacker* peut alors construire un message M' avec P' = <M', c(M')>, puis envoyer au destinataire C':

$$C' = <M', c\ (M')> \oplus RC4\ (v,k)$$

$$= P' \oplus RC4\ (v,k)$$

On remarque que le message injecté utilise le même vecteur d'initialisation v que le message original. Cependant, il est possible d'utiliser d'anciennes valeurs de v sans créer de conflit. Cela est dû au fait que le standard 802.11 [1] n'interdit pas la réutilisation d'un vecteur d'initialisation avec plusieurs paquets. Par suite, chaque récepteur est obligé d'accepter des répétitions de vecteur d'initialisation, sinon il risque d'avoir des conflits d'interopérabilité avec d'autres émetteurs.

5. Usurpation (spoofing) de l'authentification

Le mécanisme d'authentification de WEP est utilisé par les points d'accès afin d'authentifier les usagers mobiles et de leur permettre de s'associer au réseau. La séquence d'échange de messages a pour objectif de vérifier que l'usager mobile est bien celui qu'il prétend être en prouvant qu'il possède bien la bonne clé secrète.

Pour ce faire, l'usager mobile initie la communication en envoyant un message *Authentication Request* au point d'accès demandant qu'il soit authentifié. Ce dernier lui répond avec un message *Authentication Challenge* qui consiste en un message en clair que doit crypter l'usager mobile avec la bonne clé WEP et renvoyer le résultat au point d'accès via un message *Authentication Response*. L'authentification est réussie si la décryption de ce message par le point d'accès redonne le message de challenge et un *Authentication Result* est envoyé à l'usager mobile. Le fait de pouvoir générer une version cryptée du challenge prouve la possession de la bonne clé [7].

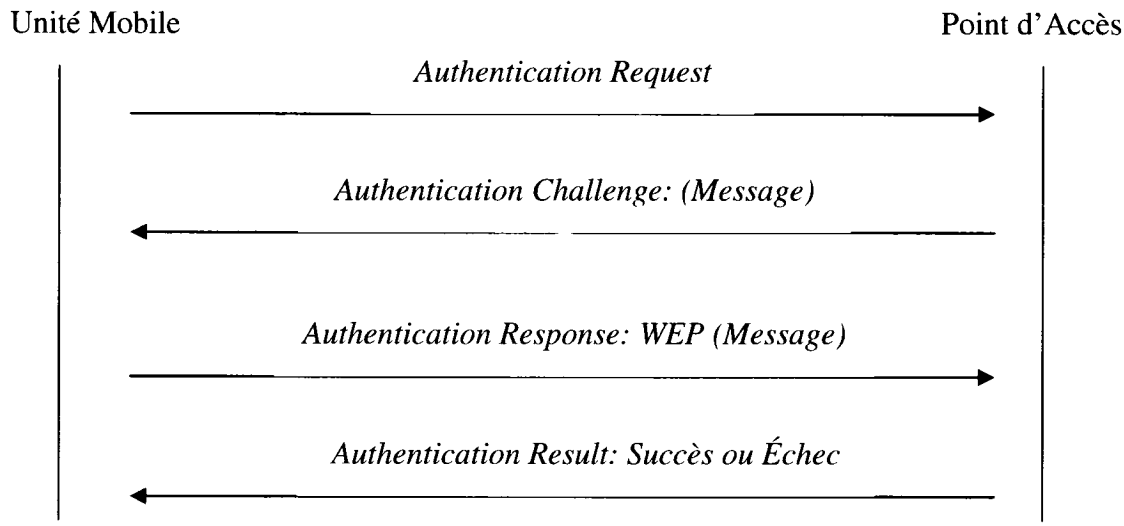

Figure 2.4 Séquence d'authentification avec WEP

En interceptant les messages 2 et 3 de la séquence d'authentification, un *hacker* peut obtenir le message du challenge M envoyé par le point d'accès PA et son

encryption avec la bonne clé envoyée par l'usager mobile U. Par conséquent, il peut retrouver la clé RC4 utilisée pour le cryptage [4] :

$$\text{Message 2: PA} \rightarrow \text{U: M}$$

$$\text{Message 3: U} \rightarrow \text{PA: <M, c (M)>} \oplus \text{RC4 (v, k)}$$

$$\text{Hacker : <M, c (M)>} \oplus \text{ <M, c (M)>} \oplus \text{RC4 (v, k)} = \text{RC4 (v, k)}$$

Une fois la clé RC4 obtenue, un *hacker* pourra crypter correctement n'importe quel message M' de challenge et par conséquent s'authentifier convenablement sans jamais avoir besoin de la bonne clé WEP.

Ce cas est encore plus grave vu qu'en pratique dans un réseau local sans fil la même clé WEP est déployée pour tout le réseau, donc un *hacker* pourra s'authentifier indéfiniment dans tout le réseau.

2.2.3 La configuration par défaut

Lors du déploiement des points d'accès, la première étape consiste à les configurer correctement. Inopportunément, beaucoup d'administrateurs réseaux gardent la configuration par défaut fournie par les manufacturiers qui renferme beaucoup de brèches de sécurité.

Tout d'abord, WEP n'est généralement pas activé. Comme nous avons vu précédemment, WEP n'est pas totalement sécuritaire mais il représente tout de même une étape en plus à franchir par un *hacker* et c'est toujours mieux que lorsque les messages sont totalement en clairs. Ensuite, certains paramètres comme le mot de passe du point d'accès, le SSID (Service Set Identifier), les paramètres SNMP (Simple Network Management Protocol), DHCP (Dynamic Host Configuration Protocol) ou encore la sélection du canal de communication sont facilement accessibles sur les sites Web des manufacturiers. Ils représentent des données critiques pour le bon fonctionnement d'un WLAN et peuvent être utilisés pour compromettre son fonctionnement [8].

D'autre part, par défaut le mode *broadcast* du SSID est activé, donc n'importe qui avec une carte réseau sans fil le reçoit du point d'accès lui-même. Par la suite, il est primordial de désactiver ce mode, sinon le changement du SSID par défaut n'a aucun intérêt.

2.2.4 L'emplacement des points d'accès

Le choix de l'emplacement physique des points d'accès représente aussi un vrai challenge. Beaucoup d'entreprises placent leurs points d'accès dans des endroits publics de leur entreprise facilement accessibles par n'importe qui, ce qui est considéré comme une erreur de planification qui met en péril la sécurité du réseau. Car une personne malveillante pourrait réinitialiser la configuration par défaut du manufacturier sans aucun problème étant donné qu'il peut atteindre physiquement le point d'accès. Ce genre d'attaque nécessite beaucoup moins de connaissances techniques mais est aussi dangereux que les autres.

Un autre point important consiste en l'installation de points d'accès par les employés eux mêmes afin d'étendre leur accès au réseau filaire de l'entreprise et sans informer les responsables du département informatique. Sachant que certains employés ne maîtrisent pas correctement la configuration des WLAN, ils peuvent inconsciemment ouvrir des brèches de sécurité dans le réseau de leur entreprise [2].

2.2.5 L'étendue du signal des points d'accès

Contrôler l'étendue du signal d'un point d'accès n'est pas chose évidente. En fait, le seul moyen de le faire avec précision consiste à parcourir les locaux de l'entreprise avec un analyseur de spectre. Etant donné que le signal émis par un point d'accès peu dépasser les frontières désirées, il est primordial de s'en assurer lors de l'installation d'un WLAN.

En effet, un type d'attaque appelé *war driving* consiste à parcourir les parkings des entreprises, avec du matériel installé dans une voiture, à la recherche d'un signal

trop puissant permettant l'accès aux ressources du réseau ou la réalisation de l'une des attaques décrites précédemment. Pour éviter ce genre de situation, un audit régulier de l'environnement du WLAN est fortement recommandé. Il permettra de savoir exactement l'étendue du signal du réseau sans fil, comment le réseau est utilisé, par quels utilisateurs et à quelle fin. Il permettra aussi de vérifier les configurations de chaque point d'accès et de leur conformité à la politique de sécurité de l'entreprise. De plus, les points d'accès installés sans l'accord du département informatique seront découverts aussi.

2.3 Analyse des améliorations à la sécurité des WLAN

Suite à la découverte des problèmes de sécurité dans les réseaux locaux sans fil, plusieurs solutions techniques, architecturales et managériales ont été proposées [9] [10] [12] [13] afin de minimiser l'impact de ces failles. Cette section présente ces principales solutions.

2.3.1 Le déploiement d'architectures sécurisées

Le recours aux architectures sécurisées a été la première méthode utilisée afin de limiter l'ampleur des dangers que pourrait créer l'installation d'un réseau local sans fil. Plusieurs techniques ont été proposées, basées principalement sur la nature critique des données à protéger, l'étendue du réseau, le montant à investir et le plan d'affaire stratégique de l'entreprise.

Par exemple, pour un réseau local sans fil personnel à domicile, il n'est pas nécessaire d'avoir recours à de telles techniques aussi pointues. Par contre, dans un milieu industriel, cela devient primordial.

Le premier point crucial consiste à déterminer si le WLAN sera une extension totale du réseau filaire et par conséquent devra permettre l'accès à toutes les ressources sans aucune restriction ou bien le WLAN sera uniquement une extension partielle du réseau filaire avec des accès sélectifs aux ressources. Dans le premier cas, un degré de

sécurité accru est préconisé vu qu'une faille minime ouvre potentiellement une porte vers le réseau tout entier qui pourrait facilement le compromettre.

Si le WLAN est uniquement une extension partielle du réseau filaire, alors deux architectures sont possibles :

1) Permettre aux usagers du WLAN un accès uniquement à la zone démilitarisée DMZ du réseau qui est généralement protégée par le *firewall* de l'entreprise. Par la suite, les usagers mobiles sont obligés de se conformer à la politique de sécurité configurée dans le *firewall* avec ce qu'elle requiert comme procédures d'authentification.

2) Le deuxième cas de figure est généralement utilisé dans les *HotSpot*, qui sont des endroits publiques comme les hôtels, les cafés ou encore les aéroports et où une connexion Internet haute vitesse sans fil est disponible à travers un WLAN. Dans ces conditions, la solution préconisée consiste à isoler complètement le WLAN du réseau interne et de lui fournir une connexion Internet dédiée à part.

Un apport important à l'efficacité des architectures sécurisées peut être assuré grâce à une bonne gestion du réseau. En effet, un monitoring assidu et régulier de ce qui se passe sur le réseau sans fil est primordial. Il permettra aux administrateurs de mieux évaluer le degré de sécurité atteint par leur installation et par conséquent de l'améliorer s'il ne répond pas aux objectifs fixés. D'autre part, la mise à jour logicielle et matérielle des points d'accès représente un point crucial au maintien d'un bon degré de sécurité. Beaucoup d'administrateurs prennent trop de temps pour installer un *patch* suite à la découverte d'une faille, ce dont profitent les *hackers*.

Par ailleurs, les plus sceptiques en matière de sécurité informatique opteront pour une stratégie complètement distincte de ce que nous avons présenté jusqu'à présent, en l'occurrence, le *Wait and See* qui consiste à interdire tout déploiement de réseaux locaux sans fil jusqu'à ce que les aspects de sécurité soient plus clairs. C'est d'ailleurs la stratégie adoptée par diverses unités du gouvernement fédéral américain, dont le Pentagone et la plupart des services militaires [9]. De plus un rapport intitulé *The*

National Strategy to Secure Cyberspace [11] recommande à toutes les agences gouvernementales américaines une précaution extrême envers les technologies sans fil.

2.3.2 Le filtrage par adresses MAC

Une autre technique utilisée afin de contrôler l'accès aux réseaux locaux sans fil est basée sur les adresses MAC (Media Access Control) des usagers mobiles. Habituellement, chaque carte réseau possède une adresse MAC unique qui permet de la distinguer des autres. D'autre part, un point d'accès peut sauvegarder une liste de contrôle d'accès contenant les adresses MAC des utilisateurs mobiles pouvant se connecter à travers lui au WLAN. Par la suite, chaque usager mobile avec une adresse MAC ne faisant pas partie de la liste de contrôle se verra automatiquement refuser l'accès. Cette technique demande un grand travail à l'administrateur réseau vu qu'il doit programmer tous les points d'accès avec les bonnes adresses et les maintenir à jour. De plus, cela limite la mobilité des utilisateurs aux points d'accès qui contiennent leurs adresses.

2.3.3 Les réseaux privés virtuels

Un réseau privé virtuel ou Virtual Private Network (VPN) est une extension d'un réseau privé comportant des liens sur des réseaux publics comme Internet. Un réseau privé virtuel sécurise une connexion en cryptant tout le trafic du réseau avant de l'envoyer sur Internet, puis en le décryptant lorsqu'il arrive à l'autre extrémité du réseau privé virtuel. Comme le réseau public transporte tout le trafic du réseau privé virtuel sous forme encapsulée, une connexion VPN est également appelée *tunnelling*.

Le VPN peut avoir recours aux protocoles PPTP (*Point-to-Point Tunneling Protocol*) ou le mode tunnel du protocole IPSec (*Internet Protocol Security*) pour gérer les tunnels et encapsuler les données privées [12].

Dans le cas d'un WLAN, des utilisateurs distants ayant accès à Internet, dans un *HotSpot* par exemple, pourront alors se connecter au réseau local de l'entreprise via un

tunnel sécurisé. L'inconvénient est qu'un VPN requière une configuration minutieuse tenant compte des problèmes d'interopérabilité et de compatibilité inter plate-forme.

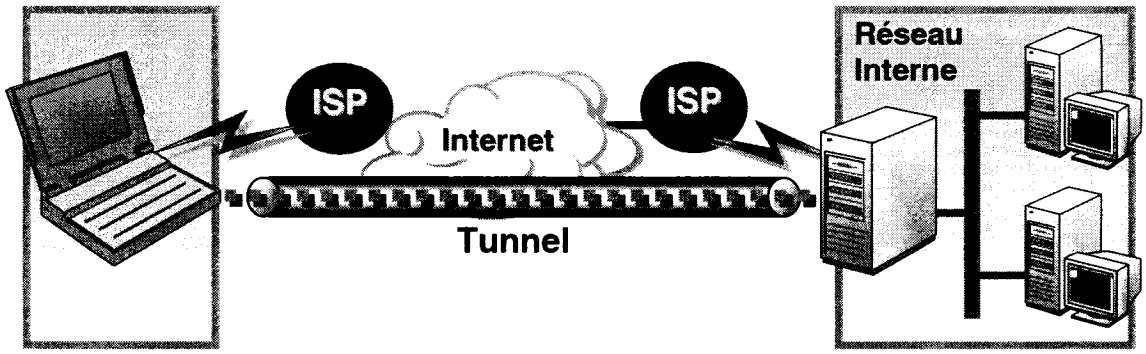

Figure 2.5 Connexion sans fil au réseau interne avec tunnel VPN

2.3.5 Le Wi-Fi Protected Access

Le *Wi-Fi Protected Access* (WPA) est prôné par la *Wi-Fi Alliance*, une organisation à but non lucratif composée des leaders industriels constructeurs de matériel pour systèmes sans fil et des entreprises fournissant des services relatifs aux réseaux locaux sans fil. Elle s'occupe principalement de certifier l'interopérabilité et la compatibilité des produits répondant aux standards IEEE 802.11 et de faire la promotion de cette technologie de réseaux locaux sans fil auprès des industriels et des clients [13].

Le besoin de sécurité étant pressant et la norme IEEE 802.11i consacrée à la sécurité des WLAN prenant trop de temps à être standardisée, la *Wi-Fi Alliance* a décidé de s'inspirer des versions préliminaires (draft) du nouveau standard afin de proposer le *Wi-Fi Protected Access* (WPA).

WPA peut être utilisé dans un environnement personnel ou professionnel. Il ne nécessite pas de mise à jour matérielle de l'infrastructure existante mais uniquement une mise à jour logicielle. WPA permet d'améliorer deux aspects importants de la sécurité sans fil, en l'occurrence, l'encryption des données et l'authentification.

Pour améliorer l'encryption WPA utilise le *Temporal Key Integrity Protocol* (TKIP) qui est basé sur une fonction de mixage de clé par paquet, un *Message Integrity Check* (MIC) et un vecteur d'initialisation étendu avec des règles de séquencement. Pour améliorer l'authentification, comme l'illustre la Figure 2.6, WPA implémente une méthode basée sur des mots de passe [13].

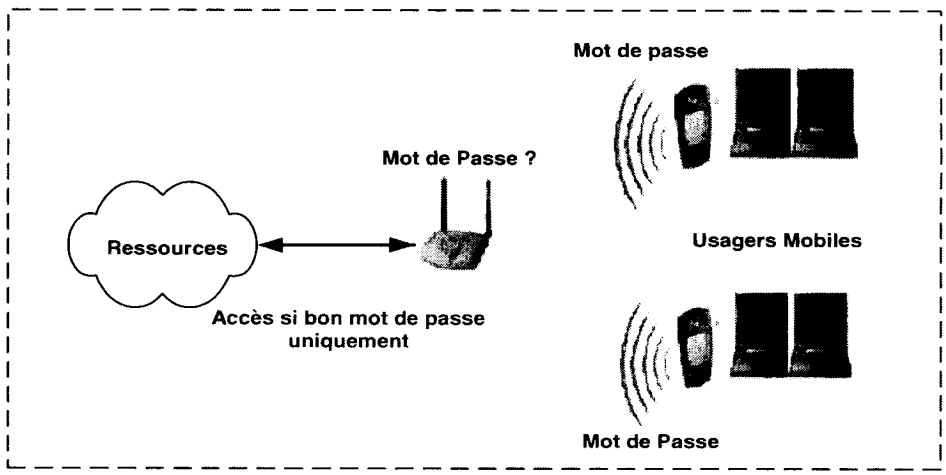

Figure 2.6 Authentification WPA

2.3.6 IEEE 802.11i

Les recommandations de l'IEEE (Institute of Electrical and Electronics Engineers) concernant toute nouvelle architecture de sécurité pour les réseaux locaux sans fil sont regroupées sous forme d'un standard IEEE 802.11i [14]. Ces recommandations sont une combinaison complexe de plusieurs protocoles distincts. 802.11i définie deux sous-classes d'architectures de sécurité : RSN (Robust Security Network) et pre-RSN (pre-Robust Security Network).

Un RSN est un réseau sécurisé qui ne permet que la création d'associations RSNA (Robust Security Network Associations). Ainsi, dans un RSN, les associations entre les stations mobiles et les points d'accès sont basées sur des dispositifs robustes faisant intervenir des mécanismes d'authentification forts. Si entre deux entités, un

niveau acceptable d'authentification n'a pas été atteint, une association ne pourra pas être créée.

Un pre-RSN est un réseau sécurisé intermédiaire permettant d'assurer la transition entre les anciennes architectures de réseaux locaux sans fil et les nouvelles architectures de type RSN. Il est composé d'éléments ne pouvant pas créer d'associations RSNA.

La sécurité dans les pre-RSN est constituée de deux sous-systèmes : l'authentification des entités IEEE 802.11 et WEP. L'authentification des entités inclut l'authentification à système ouvert (Open System Authentication) et l'authentification à clé partagée (Shared Key Authentication). L'authentification à système ouvert est un mécanisme de base constitué de deux messages et ne faisant intervenir aucune clé de cryptage. Le premier message est initié par l'unité mobile vers le point d'accès afin d'exposer son identité. Le deuxième consiste en la réponse du point d'accès avec réussite, étant donné qu'il n'y a pas d'algorithme d'authentification. En d'autres termes, l'authentification à système ouvert permet l'association de toute entité sans aucune restriction. C'est elle qui est configurée par défaut dans les points d'accès vendus par les industriels. L'authentification à clé partagée, quant à elle, fait intervenir la clé de cryptage WEP. Elle est constituée de quatre messages qui ont pour objectif de vérifier que l'unité mobile possède bien la bonne clé secrète.

La sécurité dans les RSN préconise d'autres protocoles et mécanismes. En effet, pour améliorer la sécurité du chiffrement, deux protocoles cryptographiques sont ajoutés à WEP : *Counter-Mode Cipher Block Chaining Message Authentication Code Protocol* (CCMP) et *Temporal Key Integrity Protocol* (TKIP). CCMP est basé sur l'algorithme de cryptage *Advanced Encryption Standard* (AES), alors que TKIP utilise l'algorithme de cryptage RC4. CCMP requière une mise à jour matérielle des points d'accès tandis que TKIP nécessite seulement une mise à jour logicielle. TKIP est principalement destiné à assurer l'interopérabilité entre RNS et pre-RSN. Par la suite, un réseau local sans fil pourra supporter l'utilisation simultanée de trois protocoles de chiffrement : WEP, TKIP

et CCMP. L'unité mobile et le point d'accès concerné utiliseront le plus haut degré de sécurité que les deux peuvent supporter mutuellement.

Concernant la gestion des clés, deux modes sont aussi présentés : une gestion de clés manuelle et une gestion de clés automatique. La première sollicite un administrateur réseau afin de déployer manuellement les clés de chiffrement, alors que la deuxième se base sur des mécanismes nommés « 4-Way Handshake » et « Groupe Key Handshake » afin de gérer convenablement la distribution des clés.

CHAPITRE III

MÉCANISME D'AUTHENTIFICATION PROPOSÉ

Dans le chapitre précédent, nous avons vu qu'il y avait plusieurs problèmes relatifs à la sécurité des communications dans les réseaux locaux sans fil. Nous avons décidé de nous concentrer sur l'authentification. C'est dans ce contexte que ce chapitre intervient. Il sera consacré à la spécification et à la conception d'un mécanisme d'authentification sécurisé des usagers mobiles dans les réseaux locaux sans fil. Pour ce faire, nous commencerons par une présentation des requis et des spécifications de notre mécanisme. Ensuite, nous exposerons les fondements et les principes de notre proposition. Enfin, nous terminerons par un exposé détaillé de notre conception.

3.1 Requis et spécifications

Le requis capital du mécanisme d'authentification que nous proposons est de s'assurer qu'un usager mobile est bien celui qu'il prétend être avant de lui permettre l'accès aux ressources protégées du réseau. Pour arriver à cet objectif et, par le fait même, empêcher des connexions non autorisées, notre mécanisme doit permettre un contrôle d'accès robuste. Idéalement, le contrôle devrait se faire au niveau des paquets eux-mêmes. De plus, en considérant le fait que, pour une unité mobile, tout point d'accès doit être considéré comme une entité potentiellement malveillante, un processus d'authentification mutuel devrait être envisagé.

D'autre part, les réseaux locaux sans fil ont plusieurs environnements d'utilisation, allant du réseau d'une entreprise, ou les politiques de sécurité sont très strictes, aux *Hot Spot* de fournisseurs de services Internet, disponibles dans des endroits publics et où le souci de simplicité d'utilisation prévaut sur le degré de sécurité accru. Donc, suite à ces considérations, notre mécanisme devrait permettre une assez bonne flexibilité selon son environnement d'implantation.

Un autre point important à prendre en considération est la mobilité des usagers. Notre mécanisme devrait permettre une authentification indépendamment de la localisation des utilisateurs. Le recours à une architecture trois tiers avec un serveur AAA (Authentication, Authorization, and Accounting) séparé de l'unité fournissant le service (le point d'accès) permettrait d'atteindre cet objectif.

La confidentialité des communications est aussi un point très important à considérer. Sachant que, dans un contexte sans fil, intercepter des échanges de messages ne représente ni une difficulté ni un obstacle majeur, il est primordial que notre mécanisme assure un degré de confidentialité respectable.

L'évolution des besoins étant une caractéristique évidente des réseaux locaux sans fil, il est capital de considérer l'extensibilité de notre mécanisme tout en conservant ses propriétés et capacités fonctionnelles. Certains des paramètres à considérer sont le nombre d'utilisateurs et le nombre de connexions simultanées.

Le modèle de confiance de notre mécanisme est basé sur le serveur d'authentification. Ainsi, le point d'accès et l'usager mobile doivent avoir confiance en l'intégrité du serveur puisque c'est lui qui effectue l'authentification. Par contre, le serveur d'authentification doit vérifier l'identité du point d'accès et de l'usager mobile.

Notre mécanisme est destiné à des réseaux locaux sans fil fonctionnant en mode infrastructure (Basic Service Set). La séquence exacte des messages échangés et leur contenu dépendent de la technique d'authentification utilisée. Nous abordons ces détails dans la suite du présent chapitre.

Le fonctionnement des entités de notre mécanisme doit être conforme à l'exécution d'une machine à états finis. C'est à dire que chaque entité aura une machine à états finis spécifique. L'exécution de la machine à états finis devrait permettre de vérifier la séquence de messages envoyés et le succès ou l'échec du processus d'authentification. Par suite, les machines à états finis représentent un élément crucial pour le bon fonctionnement et la sécurité du mécanisme.

3.2 Fondements et principes du mécanisme proposé

Les principes du mécanisme que nous proposons nous permettent d'apporter six contributions originales. Nous allons décrire celles-ci dans cette section.

Premièrement, nous proposons une architecture trois tiers. En effet, l'architecture du mécanisme que nous proposons, illustrée à la Figure 3.1, est constituée de trois composants: un usager mobile, un point d'accès et un serveur d'authentification. Dans les mécanismes utilisés présentement en industrie, l'authentification est basée sur une architecture deux tiers constituée uniquement de l'usager mobile et du point d'accès. Notre mécanisme permet d'introduire une couche de sécurité en plus grâce à la présence d'un troisième intervenant qui est le serveur d'authentification. Cela permet aussi d'effectuer une authentification adéquate quel que soit le point d'accès utilisé par l'usager mobile.

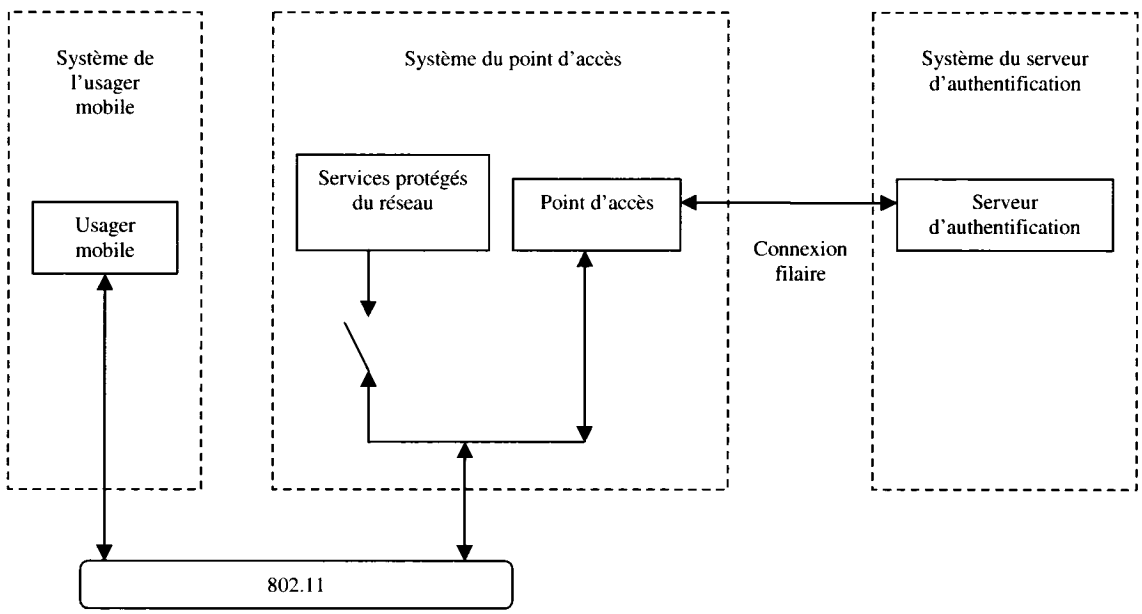

Figure 3.1 L'architecture de notre mécanisme

Deuxièmement, nous introduisons un concept innovateur d'authentification mutuelle : jusqu'à présent uniquement l'usager mobile devait prouver son identité. Dans

notre mécanisme, le serveur doit aussi prouver son identité d'où la notion d'authentification mutuelle.

Troisièmement, notre mécanisme introduit une nouvelle notion de port contrôlé. En effet, grâce à ce concept, il est possible de bloquer le trafic en cas d'échec du processus d'authentification. En fait, comme l'illustre la Figure 3.2, ce concept peut être schématisé grâce à deux ports : un port contrôlé et un port non contrôlé. Suivant l'état du port contrôlé (autorisé ou non autorisé), l'accès aux ressources du réseau peut être géré. Le port non contrôlé sert à acheminer les messages tout au long du processus d'authentification. Selon le résultat de ce processus, l'état du port contrôlé peut basculer de non autorisé à autorisé. En cas de succès de l'authentification, l'état du port contrôlé bascule vers autorisé et le trafic est acheminé via ce port. Par conséquent, l'usager mobile peut avoir accès aux ressources protégés du réseau.

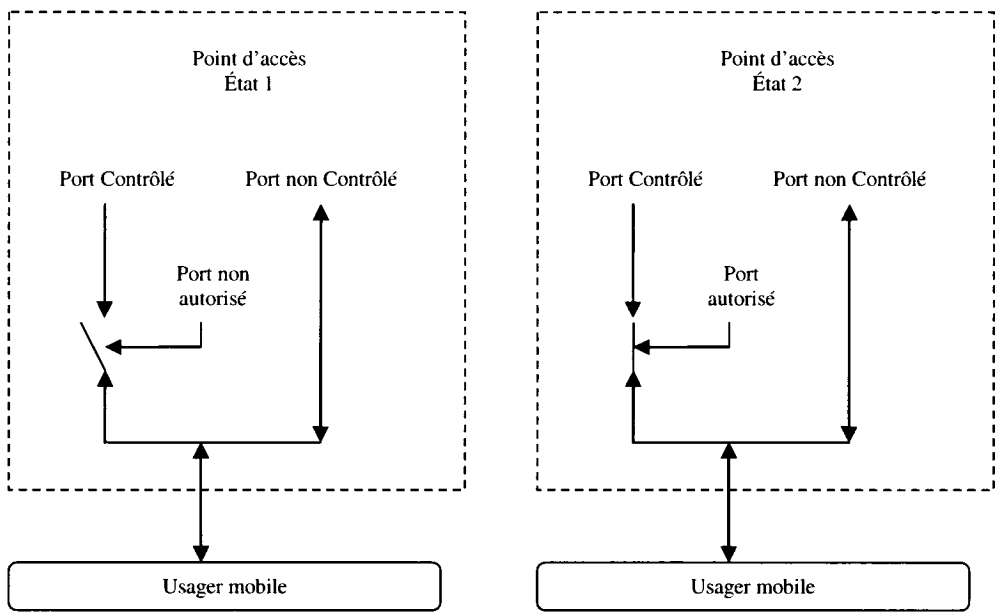

Figure 3.2 La notion de ports contrôlés dans notre mécanisme

Quatrièmement, afin d'assurer un degré de confidentialité accru, nous proposons d'utiliser un concept cryptographique très intéressant qui est les certificats numériques.

En effet, pour s'authentifier mutuellement, le serveur et l'usager mobile s'échangent leurs certificats respectifs. Afin de pouvoir véhiculer cet échange de bout en bout, c'est à dire depuis l'usager mobile jusqu'au serveur et vice versa, nous avons recours au protocole EAP (Extensible Authentication Protocol) [19]. EAP est un protocole extensible. Il permet l'utilisation de plusieurs méthodes d'authentification qui peuvent être encapsulées dans ces messages, par exemple l'authentification basée sur les cartes à puces, les mots de passe ou la biométrie. Notre mécanisme reposant sur un échange de certificats *TLS* (Transport Layer Security) est la méthode la plus adéquate. Part ailleurs, EAP est bâti sur le paradigme de communication défi-réponse (challenge-response communication paradigm). Il est composé de quatre types de messages : *EAP Request*, *EAP Response*, *EAP Success* et *EAP Failure* :

- EAP Request et EAP Response servent à encapsuler les paramètres de la méthode d'authentification utilisée ;
- EAP Success et EAP Failure permettent d'informer l'usager mobile du résultat final du processus d'authentification.

Cela dit, puisque les messages EAP doivent circuler de bout en bout, ils sont eux-mêmes encapsulés grâce au protocole EAPOL (EAP Over LAN) lors de leur transition entre le point d'accès et l'usager mobile. De même, les messages émis par l'usager mobile sont encapsulés par le point d'accès dans le bon format de trames du serveur d'authentification afin qu'ils puissent être correctement acheminés. Ici, nous supposons que tous les points d'accès communiquent avec le même serveur d'authentification. En pratique, l'authentification pourrait être distribuée sur plusieurs serveurs pour éviter un goulot d'étranglement et assurer une meilleure répartition de charge [15].

Une fois que l'usager mobile est convenablement authentifié, le port contrôlé du point d'accès passe à l'état autorisé. Par suite, les paquets provenant de l'usager mobile pourront emprunter le port contrôlé afin d'avoir accès aux ressources protégées du

réseau. La Figure 3.3 donne un aperçu simplifié et concis des protocoles utilisés dans notre mécanisme.

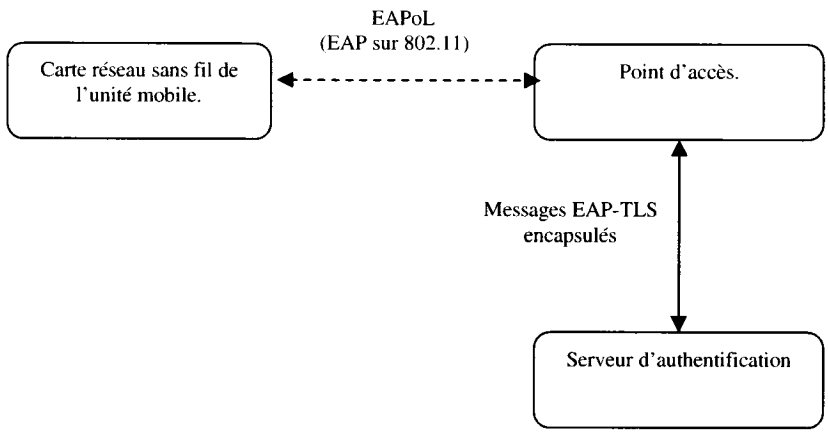

Figure 3.3 Aperçu des protocoles de notre mécanisme

Cinquièmement, le serveur d'authentification que nous proposons d'utiliser, en l'occurrence *Diameter*, constitue une contribution originale en soit. En effet, il n'a été utilisé dans aucun travail précédent relatif aux réseaux locaux sans fil. De plus, *Diameter* est le nouveau serveur AAA (Authentication, Authorization, and Accounting) prôné par l'IETF (Internet Engineering Task Force). Il est défini dans le RFC 3588 [17]. En fait, les serveurs AAA ont, à l'origine, été développé afin d'assurer des accès point à point à des serveurs de terminaux distants. Mais, avec le développement d'Internet et l'introduction de nouvelles technologies d'accès, la demande pour de meilleurs protocoles d'authentification s'est accrue. Dans notre mécanisme, *Diameter* est utilisé en mode client serveur, le rôle de client est joué par le point d'accès. Le serveur accepte les requêtes de connexion des points d'accès, authentifie les usagers mobiles et leur fournit les configurations nécessaires pour avoir accès aux services.

Par ailleurs, sachant que nous utilisons un échange de certificats transporté via le protocole EAP, l'interopérabilité de *Diameter* avec EAP est un point crucial à considérer. En effet, l'intéropérabilité entre ces deux protocoles fait intervenir deux types de messages : *Diameter-EAP-Request* et *Diameter-EAP-Answer* :

- Le message *Diameter-EAP-Request* est envoyé depuis un client *Diameter* (dans notre cas l'authentifiant, joué par le point d'accès) à un serveur *Diameter*. Selon le contenu des attributs de ce message, il sera interprété par le serveur comme un message d'initiation d'une session d'authentification ou bien un message de réponse EAP. Il peut aussi être le résultat final d'une séquence d'authentification à messages multiples.
- Le message *Diameter-EAP-Answer* est envoyé par le serveur *Diameter* au client. Selon la valeur des attributs qu'il renferme, il permet, en premier lieu, d'informer le client qu'une séquence d'authentification à messages multiples va être initiée et que le serveur attend un ou plusieurs messages de réponse. En second lieu, il permet de communiquer le résultat (succès ou échec) du processus d'authentification à l'authentifiant.

Finalement, en cas d'échec du processus d'authentification ou en cas d'attaque, notre mécanisme permet d'informer l'usager mobile de la cause de l'échec et cela, au lieu de couper directement la communication. Cela permet d'éviter que le même problème causant l'échec de l'authentification se répète plusieurs fois.

3.3 Conception du mécanisme proposé

Notre mécanisme fait intervenir trois entités distinctes : l'usager mobile noté MN, le point d'accès noté AP et le serveur d'authentification *Diameter*. Notre objectif est que l'usager mobile et le serveur d'authentification vérifient mutuellement leur identité respective afin de générer un secret partagé. Ce secret sera utilisé par la procédure de dérivation des clés afin de générer et de distribuer des clés de chiffrement qui seront utilisées pour assurer la sécurité des communications.

En se basant sur les spécifications précédentes, nous proposons un mécanisme qui peut être partagé en trois grandes phases et qui sont illustrées à la Figure 3.4.

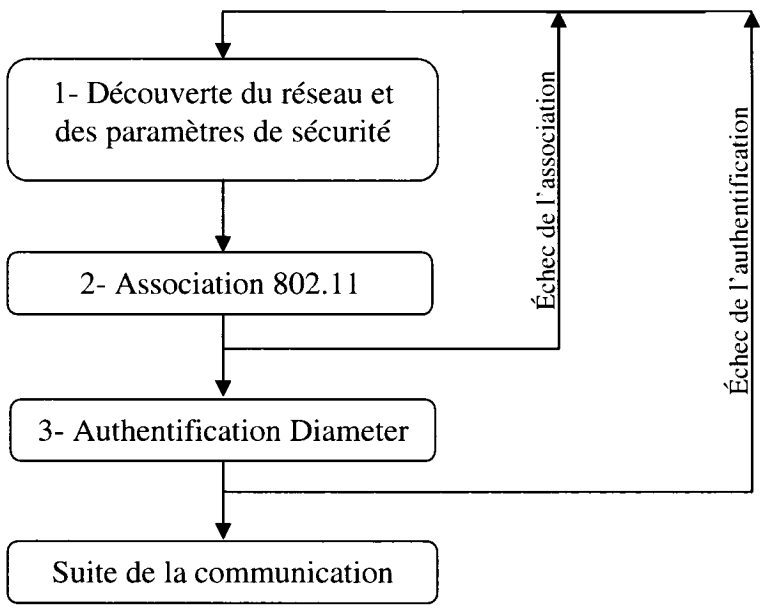

Figure 3.4 Phases du mécanisme proposé

D'un point de vue global, le mécanisme peut être vu comme suit. La première phase consiste en la découverte par un MN du réseau et des paramètres de sécurité offerts par un AP. Ensuite, la seconde phase consiste en une association 802.11 entre un MN et un AP avec négociation des paramètres de sécurité. Enfin, la dernière phase consiste en une authentification *Diameter*. En cas de panne du réseau ou d'échec de la phase d'association ou d'authentification, il y a retour au début de la phase de découverte.

D'un point de vue plus détaillé, la première étape de notre mécanisme, que nous appelons ***découverte du réseau et des paramètres de sécurité***, consiste en la découverte par une unité mobile (MN) de deux éléments : d'une part un point d'accès (AP) offrant une connectivité et d'autre part des paramètres de sécurité relatifs à ce point d'accès et requis pour cette connexion. Cette étape peut avoir lieu de deux manières :

1) le point d'accès émet périodiquement des messages *broadcast* de type *Beacon* contenant ses paramètres de sécurité ;

2) le point d'accès répond à un message *Probe Request* émis par l'unité mobile avec un message *Probe Response* contenant ses paramètres de sécurité.

Par paramètres de sécurité, nous voulons dire principalement le protocole cryptographique utilisé, c'est-à-dire WEP, TKIP ou CCMP et la technique d'authentification.

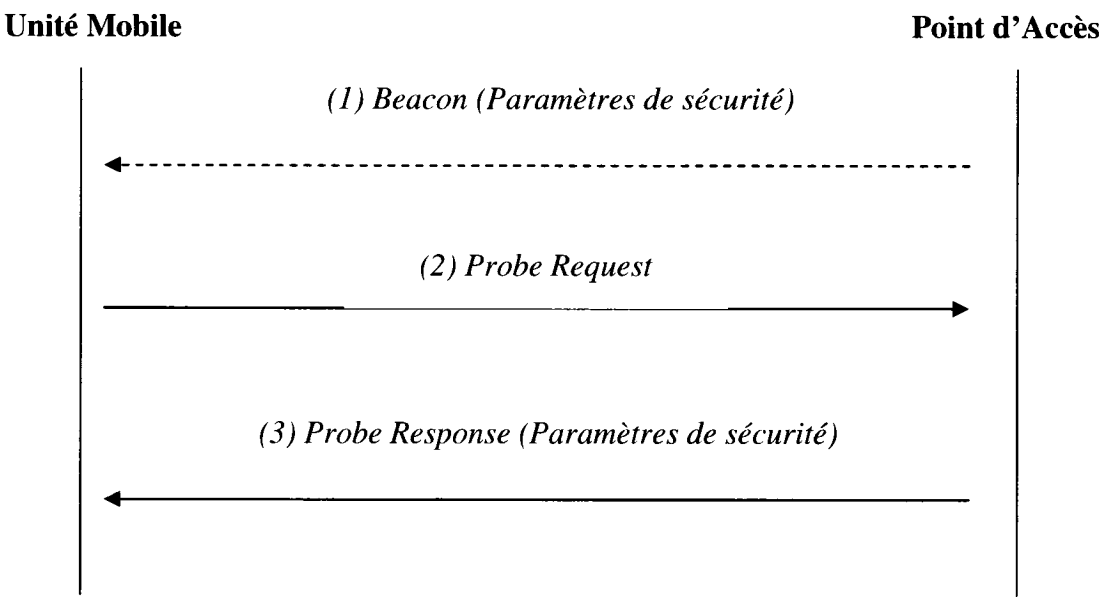

Figure 3.5 Première étape du mécanisme

La deuxième étape de notre mécanisme, que nous avons dénommée ***Association 802.11***, est basé sur le fait que la sécurité de la communication ne peut être assurée que dans un contexte d'association sécurisée. Suite à la première étape, l'unité mobile (MN) tente de s'associer à un point d'accès (AP) et cela comme suit :

1) Le MN envoie un message *Authentication Request* à l'AP.

2) L'AP répond par message *Authentication Response.*

3) Le MN envoie un message *Association Request.*

4) L'AP répond par un message *Association Response.*

Les deux messages d'authentification sont ceux de l'authentification à système ouvert, nous les utilisons pour assurer une compatibilité avec les anciens systèmes. De plus, le troisième message doit contenir les paramètres de sécurité de l'unité mobile.

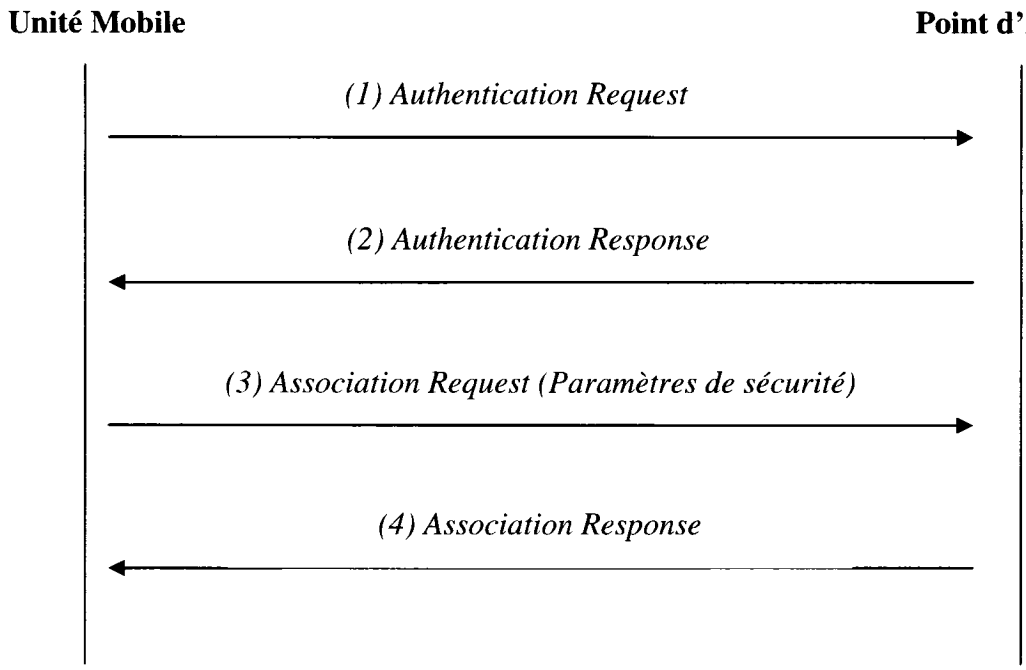

Figure 3.6 Deuxième étape du mécanisme

Authentification Diameter est la troisième étape de notre mécanisme. Elle est la plus importante aussi. Sachant que chaque association entre une unité mobile (MN) et un point d'accès (AP) crée automatiquement une paire de ports uniques, l'authentification aura lieu relativement à ces ports particuliers. Tous les messages de cette étape transitent à travers le port non contrôlé. Le port contrôlé donnant accès aux ressources du réseau reste bloqué. La séquence de messages est comme suit :

1) Le MN initie la séquence en envoyant un message *EAPOL-Start* au point d'accès AP.
2) Le point d'accès répond par un message *EAP-Request-Identity* demandant au MN de s'identifier.

3) Le MN répond avec un message *EAP-Response-Identity (ID)* contenant son identité.

4) En recevant le message *EAP-Response-Identity*, le point d'accès l'encapsule dans un message *Diameter-EAP-Request* sous forme d'un attribut *EAP Payload* et le transmet au serveur *Diameter.*

Diameter-EAP-Request
EAP-Payload (EAP-Response-Identity (ID))

5) Suite à la réception du message *Diameter-EAP-Request,* le serveur le décapsule afin d'avoir accès à l'attribut *EAP Payload,* puis initie la séquence d'échange relative à la méthode d'authentification, EAP-TLS dans notre cas. Le premier message envoyé au point d'accès est un *EAP-TLS/Start* qui est en fait un message *EAP-Request* avec *EAP-Type=EAP-TLS*, tout ce message sera encapsulé par le serveur dans un message *Diameter-EAP-Answer.*

Diameter-EAP-Answer/
EAP-Request/ EAP-Type=EAP-TLS
(EAP-TLS/Start)

6) Le point d'accès reçoit ce message, le décapsule, l'encapsule dans un message EAPOL et le transmet au MN. Le message aura la forme suivante *EAP-Request/EAP-Type=EAP-TLS (EAP-TLS/Start).*

7) Le MN répond à l'AP avec un message *EAP-Response* dont le *EAP-Type=EAP-TLS* et contenant un enregistrement *TLS client_hello.* Cet enregistrement renferme la version TLS du client, un id de session, un nombre aléatoire et l'ensemble des algorithmes de chiffrement supportés par le client.

EAP-Response/EAP-Type=EAP-TLS (TLS client_hello)

8) Le point d'accès encapsule le message reçu dans un message *Diameter-EAP-Request* sous forme d'un attribut *EAP Payload* et le transmet au serveur Diameter.

Diameter-EAP-Request
EAP-Payload (EAP-Response/EAP-Type=EAP TLS (TLSclient_hello))

9) Le serveur répond avec un message *Diameter-EAP-Answer* encapsulant un message *EAP-Request/EAP-Type=EAP-TLS* et contenant plusieurs enregistrements : *TLS serveur_hello, TLS certificate, TLS server_key_exchange, TLS certificate_request, TLS server_hello_done.*

L'enregistrement *TLS serveur_hello* renferme la version TLS du serveur, un identifiant de session, un autre nombre aléatoire et un algorithme de chiffrement.

L'identifiant de session envoyé par le serveur doit correspondre à celui reçu du le client. Ce qui indique que c'est la continuité de la session établie précédemment. L'algorithme de chiffrement est celui qu'a choisi le serveur parmi ceux proposés par le client.

Le certificat contient les informations classiques à tout certificat (nom, clé publique, signature de l'autorité de certification, date de début de validité, date de fin de validité, etc.). D'où la nécessité d'une autorité de certification ou d'une infrastructure à clé publique présente dans l'architecture du réseau d'implantation.

10) Suite à la réception de ce message, le point d'accès le décapsule, l'encapsule dans un message EAPOL et le transmet au MN. Tous les enregistrements sont gardés intacts.

11) En recevant ce message, le MN répond au point d'accès avec un message EAP-Response dont le EAP-Type=EAP-TLS et contenant les enregistrements suivants *TLS change_cipher_spec*, *TLS certificate*, *TLS certificate_verify*, *TLS client_key_exchange* et *TLS finished.*

TLS certificate représente le certificat du MN et *TLS certificate_verify* contient la réponse du MN signée avec sa clé privée.

12) Le point d'accès encapsule le message reçu dans un message *Diameter-EAP-Request* sous forme d'un attribut *EAP Payload* et le transmet au serveur *Diameter.* Il garde les mêmes enregistrements.

Le serveur *Diameter* s'occupe alors de vérifier la cohérence des informations reçues (certificat et signature numérique) afin de pouvoir accepter ou rejeter l'authentification du MN.

13) Le contenu du message suivant dépend du résultat de l'authentification. En cas d'échec, le serveur envoie un message *EAP-Request* avec l'*EAP-Type=EAP-TLS* encapsulé dans un message *Diameter-EAP-Answer* et contenant un enregistrement faisant référence à l'alerte d'erreur TLS correspondante et cela, au lieu d'interrompre la communication directement afin de permettre au MN et à l'usager de comprendre la cause de l'échec pour éventuellement la corriger. De plus, pour être sûr que le MN a bien reçu le message d'alerte, le serveur doit attendre un acquittement qui est un message *EAP-Response* vide. Par suite, il pourra envoyer un message *EAP-Failure* afin de mettre terme à la communication.

Par contre, si l'authentification réussit, le serveur répond par un message EAP-Request avec l'EAP-Type=EAP-TLS encapsulé dans un message *Diameter-EAP-Answer* et contenant les enregistrements suivants : *TLS change_cipher_spec* et *TLS finished*. Le MN lui répond alors avec un message EAP-Response vide. La séquence d'échange de message est clôturée alors par un message EAP-Success envoyé au MN.

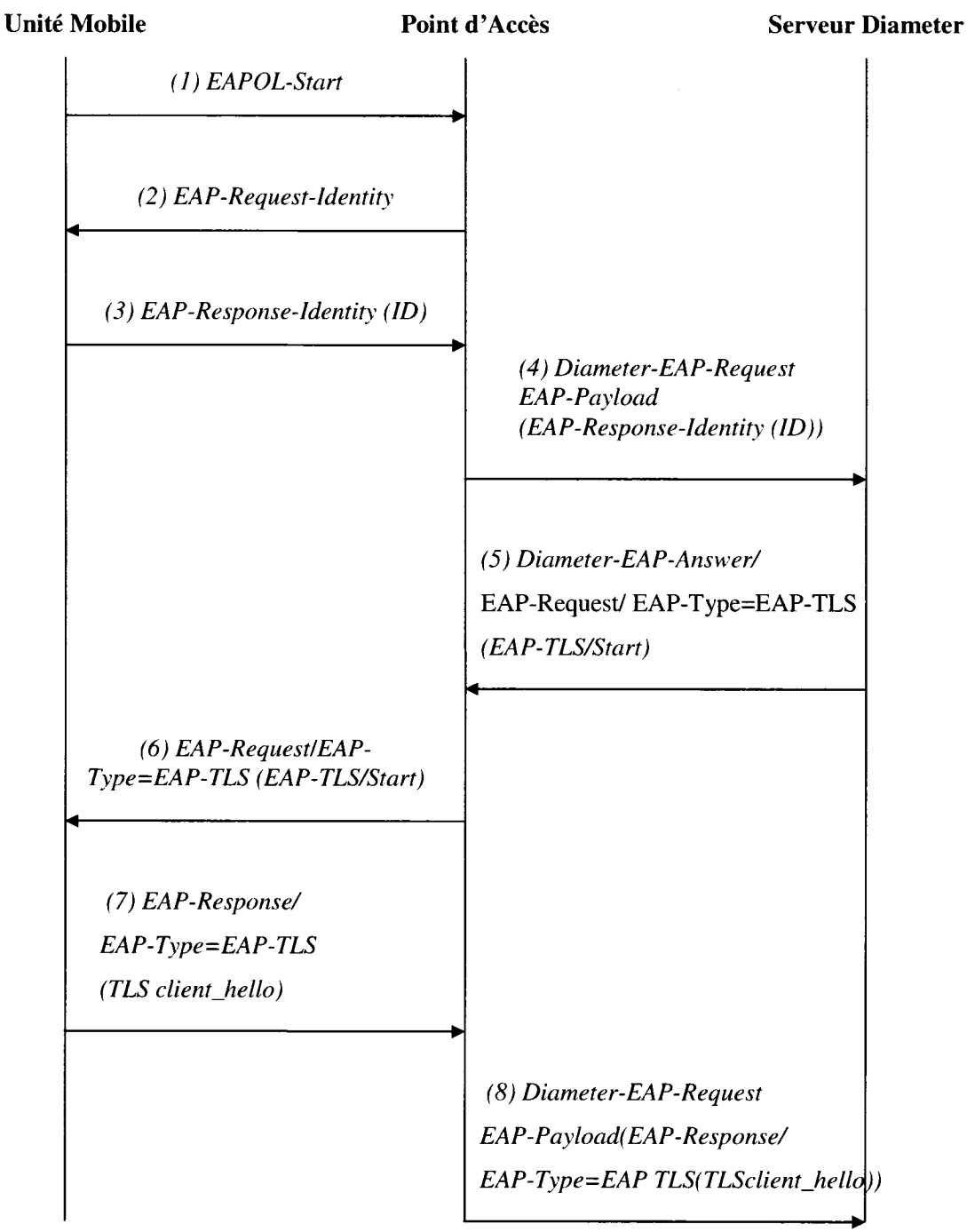

Figure 3.7 Messages 1 à 8 de l'étape 3 du mécanisme proposé

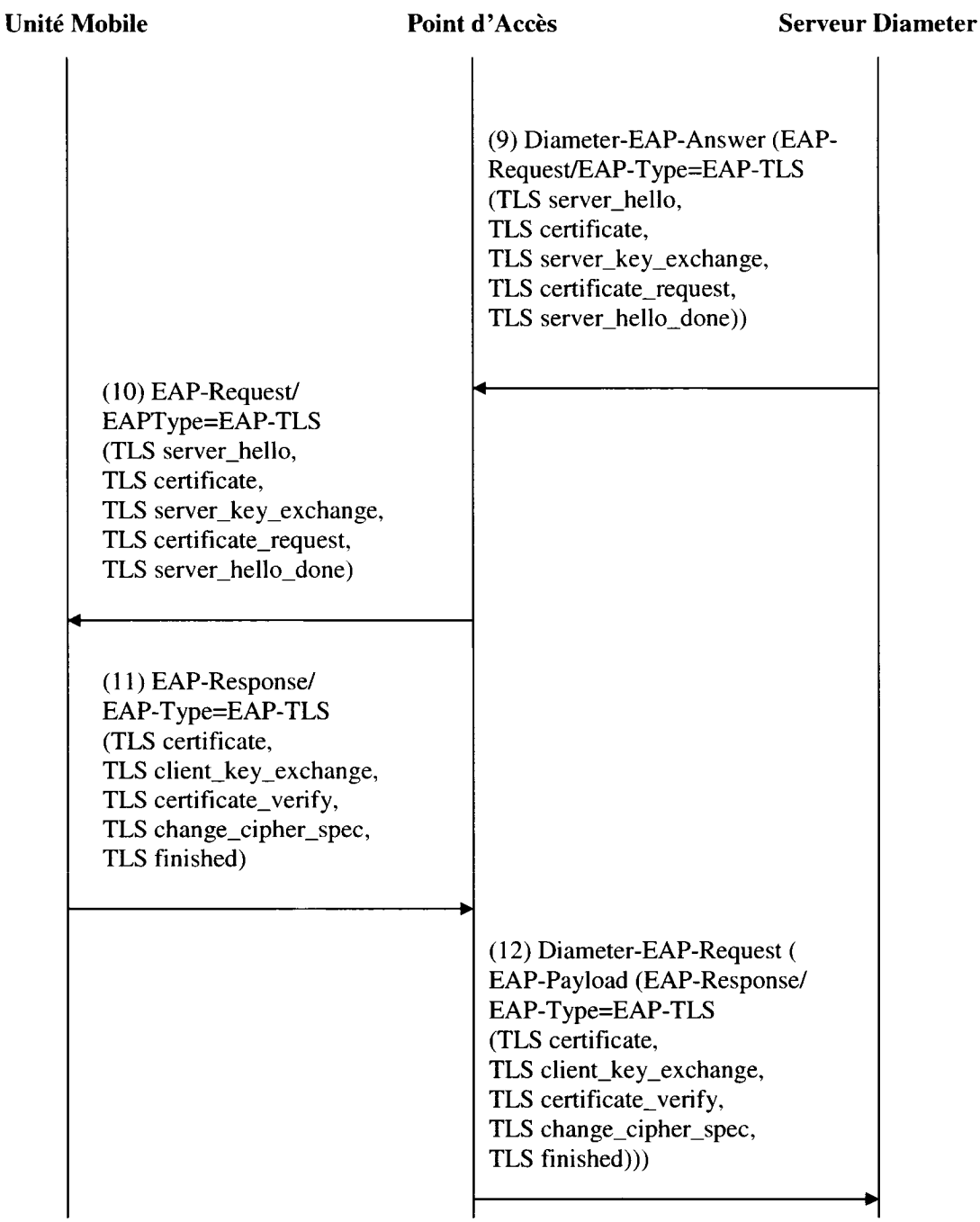

Figure 3.8 Messages 9 à 12 de l'étape 3 du mécanisme proposé

En cas d'échec de l'authentification, le processus se poursuit comme illustré à la Figure 3.9.

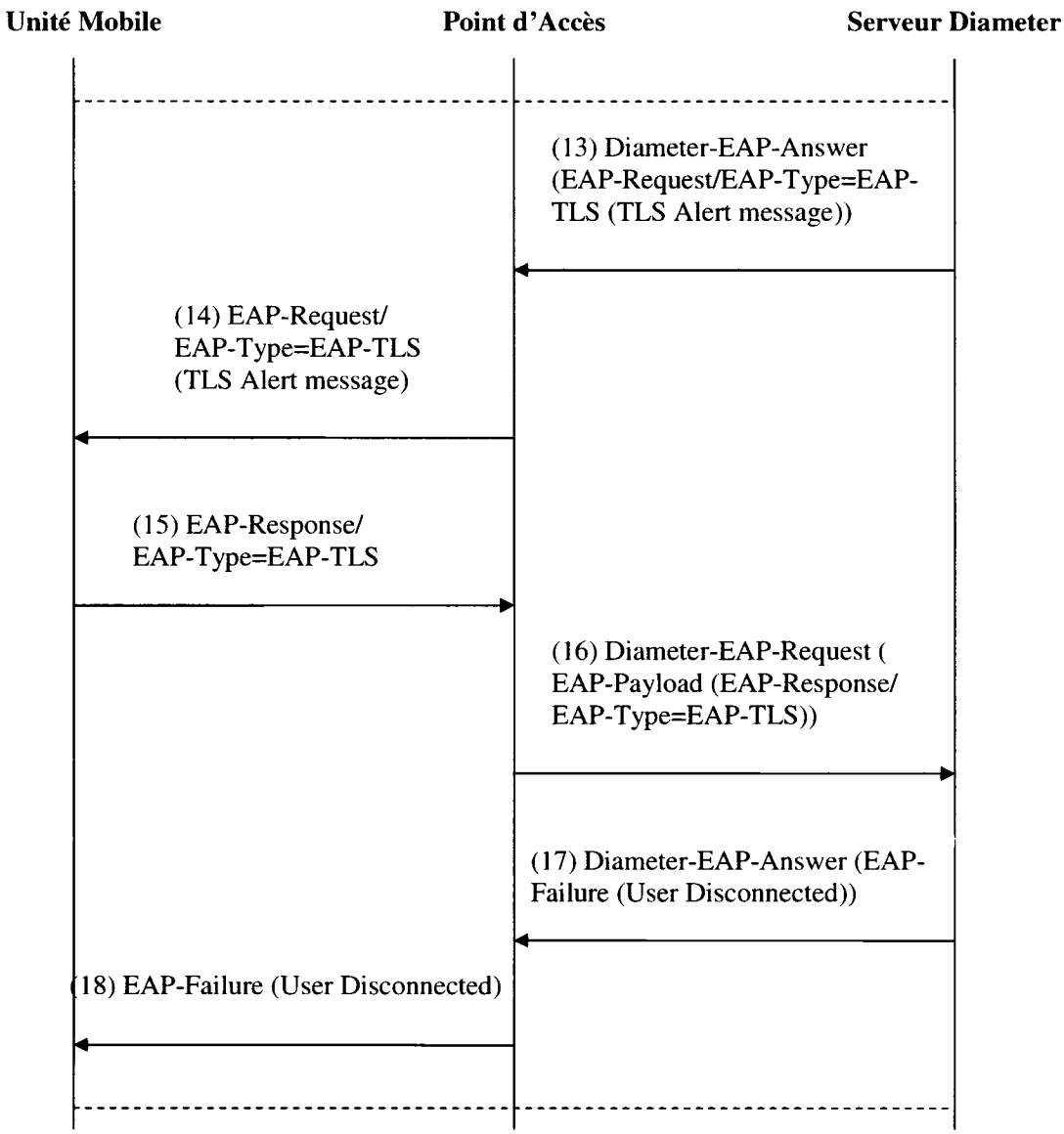

Figure 3.9 Messages 13 à 18 en cas d'échec de l'authentification

En cas de succès de l'authentification, le processus se poursuit comme illustré à la Figure 3.10.

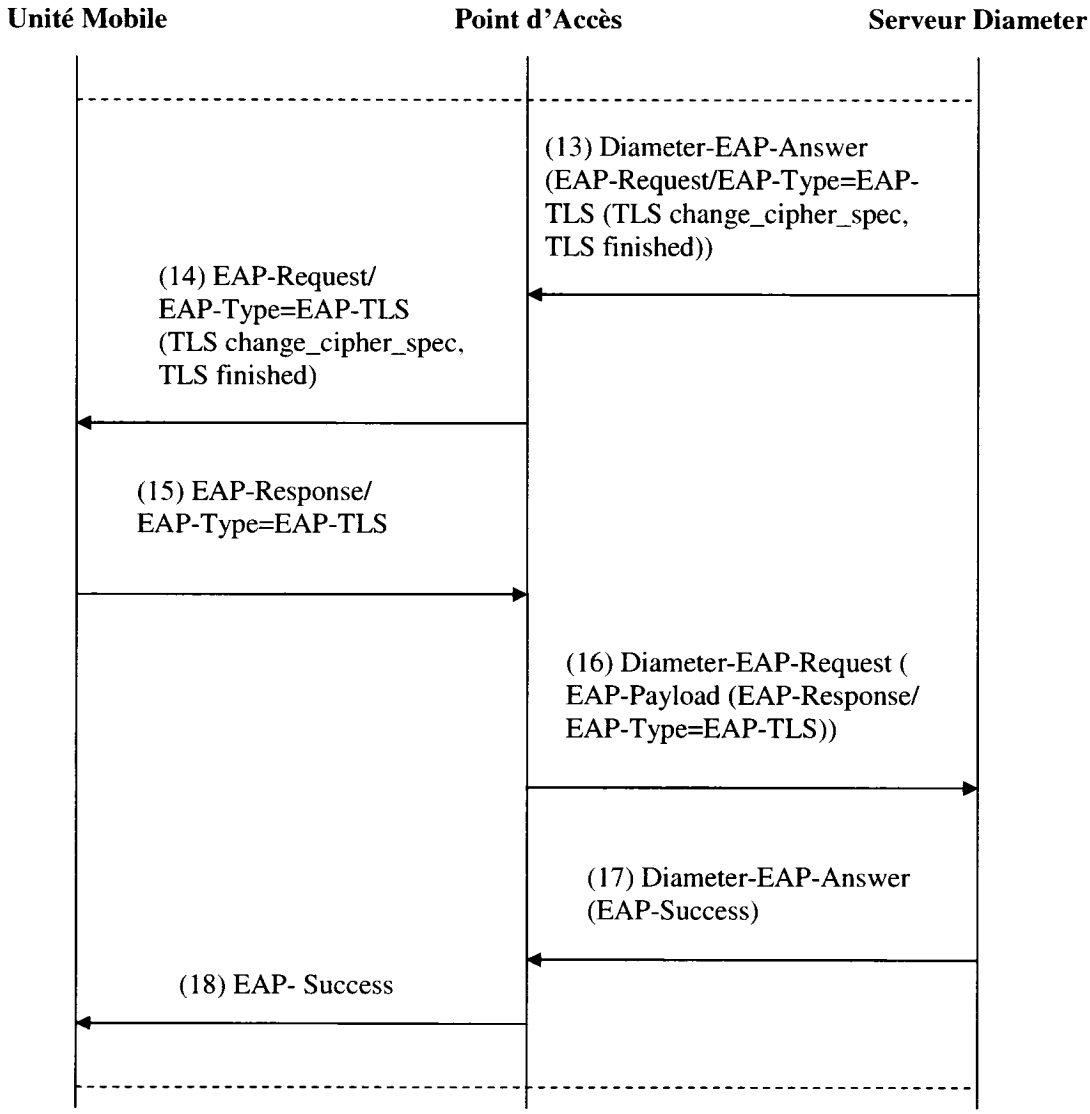

Figure 3.10 Messages 13 à 18 en cas de succès de l'authentification

La suite des étapes de la communication (distribution des clés, chiffrement des données), indépendantes de notre travail, peuvent se dérouler normalement suite au résultat positif du mécanisme d'authentification proposé.

3.4 Analyse du mécanisme proposé

Nous avons divisé notre mécanisme en trois grandes étapes distinctes et cela, afin de mieux pouvoir le concevoir. En effet, un principe connu en génie informatique consiste à diviser un gros problème assez complexe à résoudre en sous problèmes plus abordables.

Ensuite, le choix du nombre de phases a été logiquement guidé par l'objectif spécifique à atteindre par chaque étape, tout en tenant compte que l'étape suivante ne peut commencer que si l'étape précédente a correctement réussi en aboutissant à ces objectifs. Par suite, sachant que notre objectif principal est de réaliser un mécanisme sécurisé, une première étape de découverte des paramètres de sécurité des intervenants respectifs s'imposait irrévocablement. Ces paramètres nous permettaient de partir sur des bases cohérentes entre les différentes parties communicantes.

Une fois ces paramètres récupérés et nonobstant le fait qu'une authentification ne peut avoir lieu que suite à une association entre un usager mobile et un point d'accès, une étape d'association conforme à la norme 802.11, qui régit les communications dans les réseaux locaux sans fil, s'imposait automatiquement.

Enfin, puisque notre objectif ultime est de réaliser un mécanisme d'authentification, il est clair qu'une étape d'authentification sera inéluctablement présente. D'ailleurs, c'est l'étape la plus importante et la plus complexe de notre mécanisme vu qu'elle fait intervenir différents concepts cryptographiques et protocolaires. C'était une étape très délicate puisque nous avons dû résoudre plusieurs problèmes qui se posaient à nous et notamment : combien de messages faut-il utiliser ? Quel type de message ? Que faut-il mettre dans les messages et ou faut-il le mettre ?

Tout cela en assurant une cohésion globale du mécanisme et une coordination entre les différents intervenants.

Part ailleurs, lors de la conception de notre mécanisme, nous avons utilisé un serveur d'authentification *Diameter* et cela pour différentes raisons :

- Tout d'abord, nous avons été motivé par un souci d'originalité. En effet, nous sommes les premiers à proposer l'utilisation de ce genre de serveur dans un contexte de réseaux locaux sans fil.
- Ensuite, certaines études [23] [22] ont démontré qu'il est le serveur d'authentification le plus complet et qu'il répondait le mieux aux critères requis pour les serveurs d'authentification de nouvelle génération.

Finalement, concernant la technique d'authentification, nous avons opté pour une méthode basée sur les certificats, qui est certes assez complexe mais aussi la plus complète. C'est la méthode qui répondait le mieux à nos objectifs. En effet, elle offre une authentification mutuelle basée sur les certificats numériques, ce qui est primordial dans un environnement sans fil. De plus, elle permet d'éviter indéniablement les attaques de type usurpation d'identité et mascarade.

CHAPITRE IV

VALIDATION DU MÉCANISME PROPOSÉ ET RÉSULTATS

Dans le chapitre précédent, nous avons spécifié et conçu un mécanisme d'authentification sécurisé pour les réseaux locaux sans fil. Dans ce chapitre, nous allons modéliser et valider ce mécanisme. Ces deux étapes sont intimement reliées entre elles et le succès de l'une dépend inéluctablement de l'autre. Nous commencerons d'abord par la modélisation formelle à l'aide de machines à états finis. Ensuite, nous utiliserons un model-cheker pour simuler le comportement de notre mécanisme et déceler d'éventuelles erreurs de blocage ou de divergence, notamment dans le cas d'attaques. Après nous être assuré que le mécanisme fonctionne correctement, nous vérifierons qu'il répond à un ensemble de propriétés spécifiées à l'aide d'une logique formelle, ce qui permettra de confirmer sa robustesse aux attaques.

4.1 Environnement de validation

Pour réaliser les travaux relatifs à ce chapitre, nous avons utilisé l'outil UPPAAL, qui est un environnement intégré de modélisation, de simulation et de validation. C'est un produit universitaire disponible gratuitement sur Internet [26]. Il a été conçu et développé dans le cadre d'une collaboration entre le département de technologies de l'information de l'université Uppsala en Suède et le groupe de recherche en informatique fondamentale de l'université Aalborg au Danemark. Il est utilisé dans divers domaines d'application mais plus particulièrement pour les systèmes temps réel et les protocoles de communication.

UPPAAL est basé sur la notion d'automates temporisés qui sont en fait des machines à états finis munies d'horloges. Les horloges permettent de modéliser le temps qui est considéré comme une notion globale et qui évolue à la même cadence pour tout le système. D'autre part, il est permis de tester la valeur d'une horloge ou de la

réinitialiser. UPPAAL est composé de trois modules : une interface graphique, un simulateur et un model-checker.

L'interface graphique permet la modélisation du système sous forme de machines à états finis. **Le simulateur** permet à l'usager d'examiner de façon interactive le comportement du système à travers une séquence de changements d'états. Par la suite, il permet une détection rapide des erreurs de modélisation. Par ailleurs, il permet de visualiser graphiquement une trace d'exécution pour que l'utilisateur puisse suivre étape par étape l'évolution du système.

Le model-checker permet d'explorer tous les comportements possibles du système étudié. Il est conçu pour vérifier certaines propriétés, comme par exemple celles concernant l'accessibilité des états, le non blocage et la vérification des contraintes sur les variables. Il prend en entrée deux arguments : le réseau d'automates et une expression de la propriété à vérifier. Il retourne comme résultat que la propriété est satisfaite ou non. Il génère aussi une trace de diagnostic permettant de « déboguer » le système en cas d'erreur.

Un système dans UPPAAL est composé de processus concurrents. Chaque processus est modélisé grâce à un automate. Chaque automate a un ensemble d'états et des transitions permettent de passer d'un état à un autre. Afin de contrôler quand une transition est déclenchée, on utilise des « gards » qui sont des conditions sur des variables ou sur des horloges et qui définissent quand une transition est activée. De plus, un ensemble d'actions peuvent être programmées avec une transition, comme par exemple la mise à jour d'une variable ou la réinitialisation d'une horloge. Par ailleurs, afin d'assurer une cohérence entre les différents processus, on utilise des canaux de synchronisation, par exemple dans le cas ou deux processus transitent vers de nouveaux états simultanément.

4.2 Modélisation du mécanisme proposé

La modélisation du mécanisme vise à décrire clairement et sans ambiguïté, au moyen d'un modèle formel, le comportement du mécanisme. Cette étape nous permettra, par la suite, de programmer le modèle conçu en utilisant un « model-cheker » afin de simuler l'interaction entre les différents intervenants de notre mécanisme.

La modélisation doit être rigoureuse, concise et précise pour permettre une meilleure compréhension et pour pouvoir étudier tous les comportements possibles du système. Afin de répondre à ces besoins, différents types de modèles existent (automate fini ou machine à états finis, automate temporisé, réseau de Pétri, algèbres de processus...).

Dans notre cas, nous avons utilisé une modélisation sous forme de machines à états finis. Une machine à états finis représente le fonctionnement et les opérations effectuées par un intervenant sous forme d'états connectés mais mutuellement exclusifs. Les transitions entre ces différents états définissent le comportement du mécanisme. Cependant, un état unique de la machine peut être actif à un instant donné.

La difficulté majeure consiste à trouver comment passer d'un mécanisme faisant intervenir différentes séquences de messages en un ensemble d'automates, sachant qu'il est crucial de conserver une équivalence entre le modèle conçu et la réalité. D'autre part, il est primordial aussi d'assurer une cohérence du système global et une synchronisation entre les différents intervenants, qui sont en l'occurrence: l'usager mobile, le point d'accès et le serveur d'authentification *Diameter*. Pour y arriver, nous avons fait une analyse par énumération. Cette analyse consiste, dans un premier temps, à définir un ensemble d'états par lesquels passent les différents intervenants de notre mécanisme et dans un deuxième temps, d'utiliser des canaux de communication afin d'en assurer la synchronisation. Comme résultats de cette étape, les états suivants ont été répertoriés :

- Pour le point d'accès, les différents états sont : Initialisation, En_connexion, En_authentification, Authentification_réussie, Echec_authentification ;
- Pour l'usager mobile, les différents états sont : Initialisation, Déconnecté, En_connexion, En_authentification, Authentifié, Non_authentifié ;
- Pour le serveur d'authentification *Diameter*, les différents états sont : Initialisation, Succès, Echec, Attente.

Tout d'abord, les différents acteurs de notre mécanisme seront identifiés à des processus. Ensuite, pour modéliser le comportement global du système, nous représenterons chaque processus sous forme d'une machine à états finis. Dans la suite, nous adopterons la notation suivante : l'usager mobile sera modélisé par le *processus MN ( )*, le point d'accès par le *processus AP ( )*, et le serveur d'authentification *Diameter* par le *processus Srv_Diameter ( )*. D'autre part, afin de simuler le comportement de notre mécanisme en cas d'attaques, nous avons pensé à modéliser un usager mobile malhonnête et un point d'accès malicieux respectivement nommés *processus UM ( )* et *processus PAM ( )*.

Par ailleurs, nous rappelons qu'à chaque instant, le système se trouve dans une disposition bien précise caractérisée par un vecteur qui donne l'état où se trouve chaque processus et ses évolutions possibles lors de la prochaine étape.

Le processus MN ( ) : Son modèle est présenté à la Figure 4.1. C'est le processus responsable de déclencher le mécanisme. Il débute à l'état *Initialisation*, depuis lequel il envoie un message *start* au processus *AP ( )* afin de l'informer du début de la phase d'authentification et transite automatiquement vers l'état *Déconnecté*. Il reste dans cet état jusqu'à la réception du message *request_id* du processus *AP ( )*, qui lui demande son identifiant et le fait transiter vers l'état *En_connection*. Il y répond alors avec le message *response_id*, contenant son identifiant et transite vers l'état *Connecté*.

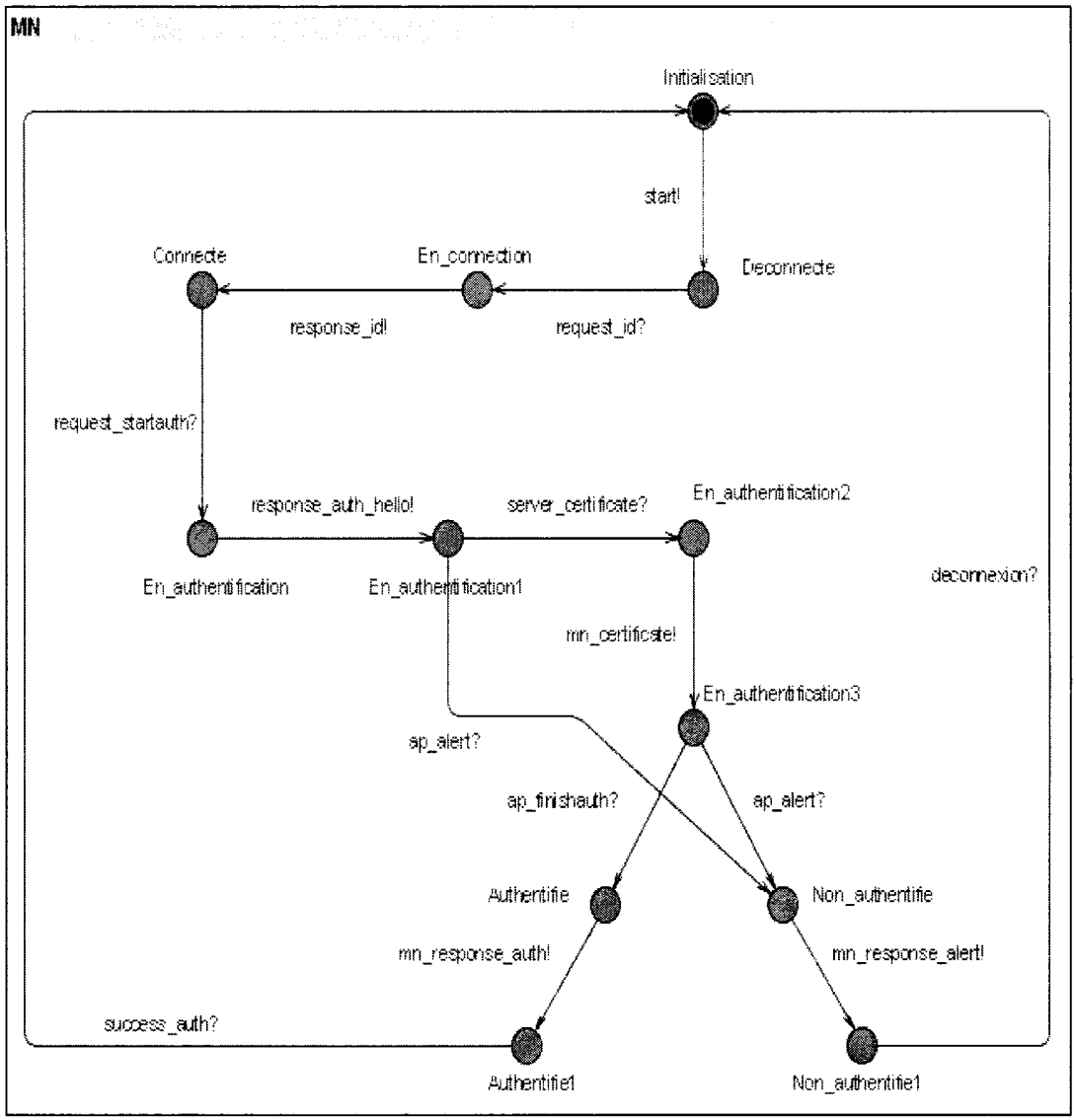

Figure 4.1 Modèle du processus MN

Il ne pourra franchir cet état que s'il reçoit un message *request_startauth* du processus *AP ( )* qui marque le début effectif de l'étape d'authentification et par conséquent le fera passer vers l'état *En_authentification*. Cet état est subdivisé en trois sous-états nommés *En_authentification1, En_authentification2,* et *En_authentification3*. En fait, ces états représentent les étapes d'échange de certificats entre l'usager mobile et le serveur d'authentification *Diameter*. La première transition entre l'état

En_authentification et le sous-état *En_authentification1* a lieu suite à l'envoi par le processus *MN ( )* du message *response_auth_hello* au processus *AP ( )* afin de l'informer qu'il est prêt à recevoir le certificat du serveur d'authentification. Par la suite, une transition de cet état n'aura lieu que suite à la réception par le processus *MN ( )* du message *server_certificate* qui le fera passer vers l'état *En_authentification2*. Cela dit, en cas de détection d'attaque, le processus *MN ( )* pourrait recevoir un message *ap_alert* qui le ferait transiter directement vers l'état *Non_authentifié*. Dans le cas où il n'y a pas eu d'attaque, après avoir reçu le certificat du serveur, c'est l'usager mobile qui doit envoyer son certificat grâce au message *mn_certificate*, puis il transite vers l'état *En_authentification3*. Arrivé à ce stade, le processus *MN ( )* attend le résultat de l'authentification qui peut être positif ou négatif. Par la suite, depuis cet état, deux transitions sont possibles selon le message qui sera reçu par le processus *MN ( )* :

1) Si le message reçu est *ap_finishauth*, alors cela veut dire que la phase d'authentification touche à sa fin et que l'usager mobile s'est correctement authentifié : le processus transite vers l'état *Authentifié*. Cependant, le processus *MN ( )* doit confirmer qu'il a bien reçu ce message en envoyant le message *mn_response_auth* au processus *AP ( )*. Ce message fait état d'acquittement et lui permet de transiter vers un sous-état *Authentifié1* où il attend le dernier message de validation de la réussite de l'authentification, c'est-à-dire *success_auth* qu'il doit recevoir du processus *AP ( )*. La réception de ce message signifie la clôture définitive de la phase d'authentification et le processus *MN ( )* revient au début de la machine à états finis, plus exactement à l'état *Initialisation*.

2) Si le message reçu est *ap_alert*, alors cela signifie que l'authentification a échoué et l'usager mobile est informé de la cause de l'échec grâce à ce message. Le processus *MN ( )* transite alors vers l'état *Non_authentifié*, il doit cependant confirmer qu'il a bien reçu ce message en envoyant au processus *AP ( )* un message *mn_response_alert* qui est considéré comme

un accusé de réception. Il transite alors vers le sous-état *Non_authentifié1*. La connexion sera totalement interrompue une fois que le processus *MN ( )* aura reçu le message *déconnexion* qui réinitialisera sa machine à états finis avec une transition vers l'état *Initialisation*.

Le processus AP ( ) : Son modèle est présenté à la Figure 4.2. Ce processus est plus complexe à modéliser parce qu'il est en interaction directe et simultanée avec les processus *MN ( )* et *Srv_Diameter ( )*. Il débute à l'état *Initialisation* qu'il ne pourra quitter que suite à la réception d'un message *start* envoyé par le processus *MN ( )* qui le fera transiter vers l'état *En_connexion*. Afin d'assurer une meilleure modélisation de l'interdépendance avec les processus *MN ( )* et *Srv_Diameter ( )*, cet état est subdivisé en trois sous-états (*En_connexion1*, *En_connexion2* et *En_connexion3*) qui modélisent tous une phase de connexion durant laquelle le processus *AP ( )* négocie avec les deux autres processus. En effet, il commence par envoyer un message *request_id* au processus *MN ()* afin de lui demander de s'identifier et transite vers l'état *En_connexion1* où il attend la réponse du processus *MN ( )* c'est à dire un message *response_id* qui lui permettra de transiter vers l'état *En_connexion2*. Par la suite, c'est la phase de communication avec le processus *Srv_Diameter ( )* qui sera entamée et l'identifiant reçu lui sera transmis sous forme d'un message *diameter_request_id* qui permettra de faire transiter le processus *AP( )* vers l'état *En_connexion3*. Arrivé à ce stade, le processus *AP ( )* devra attendre la réponse du processus *Srv_Diameter ( )* qui consiste en un message *diameter_answer_startauth*. Ce message marquera le début de la phase d'authentification avec une transition vers l'état *En_authentification* du processus *AP ( )*.

De même que précédemment et pour les mêmes raisons, cet état est subdivisé en plusieurs sous-états. Tout d'abord, le processus *AP ( )* doit envoyer un message *request_startauth* au processus *MN ( )* afin de l'informer du début de la phase principale d'authentification, il transite alors vers l'état *En_authentification1* où il devra attendre la confirmation de réception du processus *MN ( )* qui doit être sous forme d'un message *response_auth_hello*.

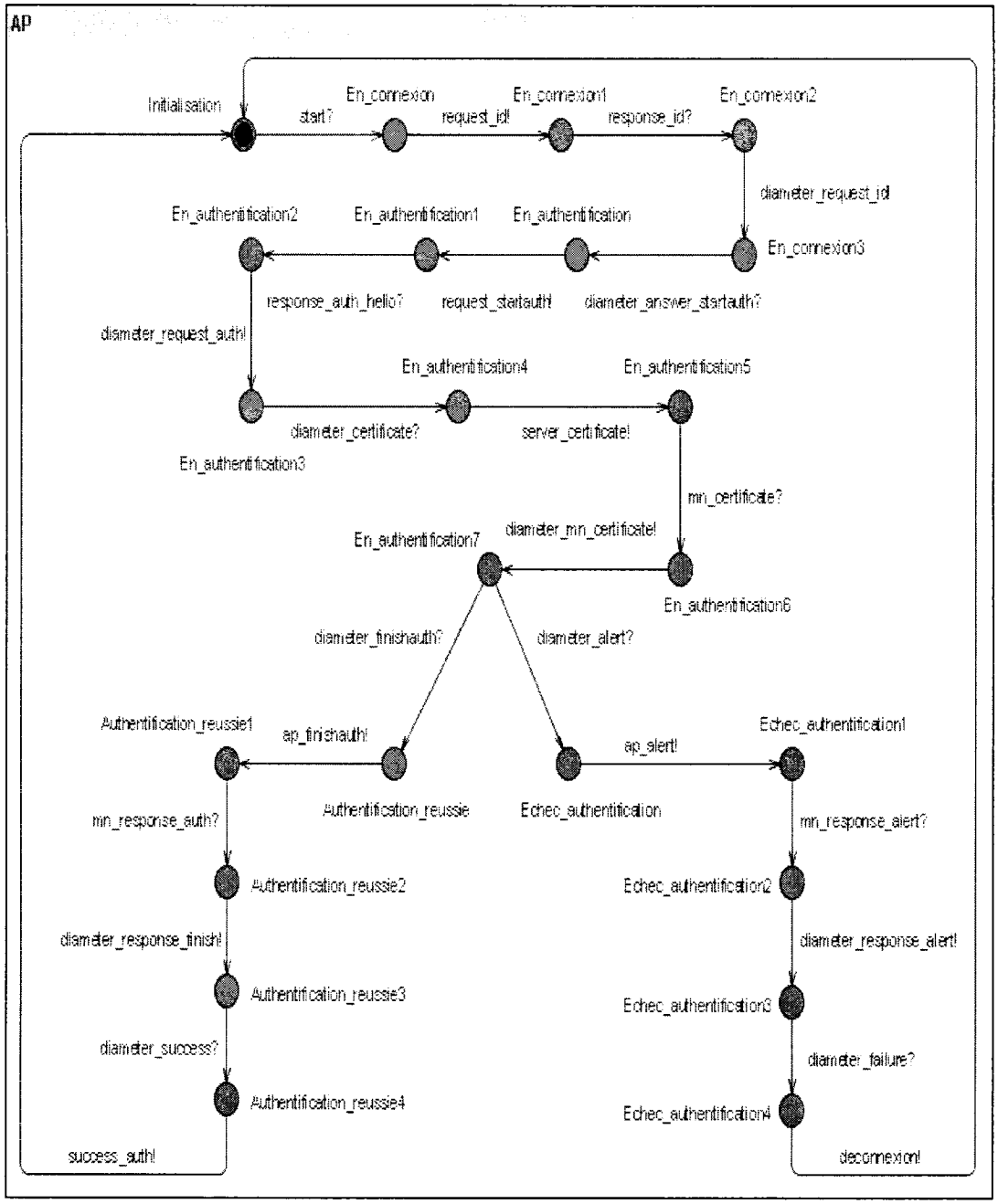

Figure 4.2 Modèle du processus AP

Une fois ce message reçu, il pourra transiter vers l'état *En_authentification2*. À partir de ce stade, c'est l'étape d'échange de certificat qui débute. En effet, le processus

AP ( ) envoie une demande de certificat au processus *Srv_Diameter ( )* sous forme d'une requête *diameter_request_auth* et transite vers l'état *En_authentification3* où il devra attendre la réponse du processus *Srv_Diameter ( )* qui doit être un message *diameter_certificate* contenant le certificat du serveur. Une fois ce message reçu, le processus *AP ( )* pourra transiter vers l'état *En_authentification4*. La prochaine étape consiste à transmettre le certificat reçu au processus *MN ( )* et cela, grâce au message *server_certificate* qui permet par la même occasion de transiter le processus *AP ( )* vers l'état *En_authentification5*. À ce moment, c'est au tour du processus *MN ( )* d'envoyer son certificat. Par la suite, le processus *AP ( )* ne pourra quitter le dernier état atteint que suite à la réception du message *mn_certificate* qui lui permettra de transiter vers l'état *En_authentification6*. Il devra ensuite, transmettre le certificat reçu au processus *Srv_Diameter ( )* et cela grâce au message *diameter_mn_certificate* qui lui permettra de transiter vers l'état *En_authentification7* qui constitue la dernière étape avant le résultat préliminaire de l'authentification. En effet, arrivé à cette phase le processus *AP ( )* devra attendre dans cet état jusqu'à la réception du résultat de l'authentification émis par le processus *Srv_Diameter ( )* qui, selon la situation, permettra de transiter vers les deux nouveaux états suivants :

1) Si le message reçu est *diameter_finishauth*, cela veut dire que l'authentification a réussi et le processus *AP ( )* transite vers l'état *Authentification_réussie*. Il doit alors informer le processus *MN ( )* en lui envoyant un message *ap_finishauth* qui permet au processus *AP ( )* de transiter vers le sous-état *Authentification_réussie1*. Par la suite, il devra rester dans cet état jusqu'à la réception de la confirmation du processus *MN ( )* sous la forme d'un message *mn_response_auth*, qui lui permettra de passer à l'état *Authentification_réussie2*. À ce moment, le processus *AP ( )* doit transmettre cette confirmation au processus *Srv_Diameter ( )* et cela, grâce au message *diameter_response_finish*, il pourra par la même occasion transiter vers l'état *Authentification_réussie3*. Arrivé à ce stade, le processus *AP ( )* restera dans cet état et devra attendre le

message de réussite final provenant du processus *Srv_Diameter ( )* et nommé *diameter_success* qui, une fois reçu, lui permettra de passer à l'état *Authentification_réussie4*. L'étape ultime est alors atteinte, il ne reste plus au processus *AP ( )* qu'à envoyer un message *success_auth* au processus *MN ( )* afin de l'informer de la réussite finale de cette session d'authentification. Une fois ce message envoyé, la machine à états finis du processus *AP ( )* sera automatiquement réinitialisée avec une transition directe vers l'état de départ, c'est-à-dire *Initialisation*.

2) Si le message reçu est *diameter_alert*, alors il y a eu un problème et l'authentification a échoué. Par la suite, le processus *AP ( )* transite vers l'état *Echec_authentification*. Pour les mêmes raisons que précédemment, cet état est subdivisé en plusieurs sous-états. Par ailleurs, avant d'interrompre la communication, le processus *AP ( )* doit informer le processus *MN ( )* de la cause de l'échec et cela, grâce au message *ap_alert* qui lui permet de transiter vers l'état *Echec_authentification1* où il devra attendre un acquittement de réception de la part du processus *MN ( )* concrétisé par un message *mn_response_alert*. Ce message fera transiter le processus *AP ( )* vers l'état *Echec_authentification2*. À ce moment, cet acquittement reçu doit être acheminé au processus *Srv_Diameter ( )* grâce à un message *diameter_response_alert*, qui permettra au processus *AP ( )* d'arriver à l'état *Echec_authentification3*. Il ne reste plus qu'a attendre le message d'échec final provenant du processus *Srv_Diameter ( )* dénommé *diameter_failure* qui fera transiter le processus *AP ( )* vers le dernier état, en l'occurrence *Echec_authentification4*. Finalement, le processus *AP ( )* peut interrompre la communication en envoyant un message *déconnexion* au processus *MN ( )* qui, par la même occasion, permettra de réinitialiser sa machine à états finis avec une transition vers l'état *Initialisation*.

Le processus Srv_Diameter ( ) : son modèle est présenté à la Figure 4.3. Ce processus débute à l'état *Initialisation*, il sera déclanché suite à la réception de la requête *diameter_request_id* provenant du processus *AP ( )*, qui en fait renferme l'identifiant de l'usager mobile et demande l'ouverture d'une session d'authentification. Cette requête fait transiter le processus *Srv_Diameter ( )* vers le sous-état *Initialisation1*. Une fois la session d'authentification initialisée, le processus *Srv_Diameter ( )* doit en informer l'usager mobile en envoyant un message *diameter_answer_startauth* au processus *AP ()*. Une fois ce message envoyé, le processus *Srv_Diameter ( )* transite vers l'état *En_attente*. Il reste dans cet état jusqu'à la réception du message *diameter_request_auth* du processus *AP ( )*, qui lui demande de s'authentifier. Étant donné que nous avons utilisé une technique d'authentification mutuelle, le serveur doit aussi s'authentifier à l'usager mobile. Par la suite, une fois ce dernier message reçu, le processus *Srv_Diameter ( )* transite vers l'état *En_authentification*. Il faut remarquer aussi que si le serveur avait fait l'objet d'une attaque d'un point d'accès malicieux, elle est détectée à l'état *En_attente*, suite à la réception par le processus *Srv_Diameter ( )* du message *apm_diameter_request_auth*, ce qui le fait transiter directement vers l'état *En_authentification2* et la variable *apm* est affectée à 1 afin d'indiquer qu'il s'agit d'une attaque d'un point d'accès malicieux.

Dans la situation sans attaque, à l'état *En_authentification*, le serveur envoie son certificat dans le message *diameter_certificate* au processus *AP ( )*. Par la suite, il transite vers le sous-état *En_authentification1*. C'est alors au tour de l'usager mobile de s'authentifier et le processus *Srv_Diameter ( )* restera dans son dernier état, jusqu'à la réception du message *diameter_mn_certificate* envoyé par le processus *AP ( )* et contenant le certificat de l'usager mobile. Une fois ce message reçu, le processus *Srv_Diameter ( )* transite vers le sous-état *En_authentification2* et affecte la variable *apm* à 0 afin d'indiquer qu'il n'y a pas eu d'attaque d'un point d'accès malicieux. D'autre part, dans le cas ou le serveur aurait fait l'objet d'une attaque d'usurpation d'identité, elle est détectée à ce stade suite à la réception du message *diameter_um_certificate*. Le processus *Srv_diameter ( )* transite alors vers le sous-état

En_authentification2 et affecte la variable *cert* à 0 afin d'indiquer que le certificat reçu est erroné.

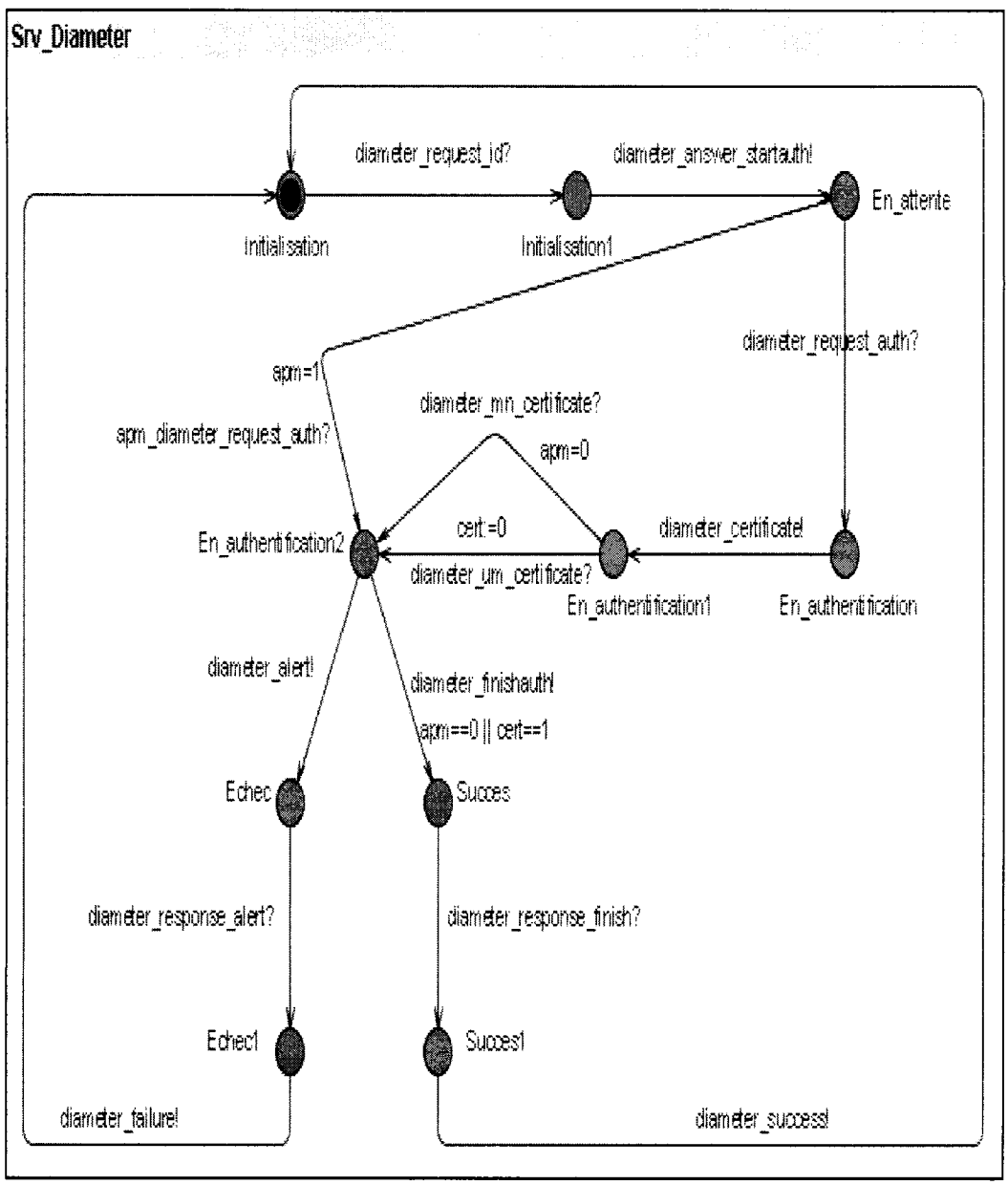

Figure 4.3 Modèle du processus Srv_Diameter

Arrivé à cette étape et selon le contenu des variables représentant l'état du système et des certificats reçu, le serveur *Diameter* doit décider du résultat de la phase d'authentification ; par la suite, deux transitions seront possibles pour le processus *Srv_Diameter ( )* :

1) Si l'authentification est positive et que le système n'a pas fait l'objet d'attaques, c'est-à-dire que la variable *apm* a la valeur 0 ou la variable *cert* a la valeur 1, le processus *Srv_Diameter ( )* envoie le message *diameter_finishauth* au processus *AP ( )*, afin de l'informer que l'authentification a réussi et transite vers l'état *Succès*. Cependant, le processus *Srv_Diameter ( )* doit attendre une confirmation que l'usager mobile a bien reçu ce message. Cette confirmation sera émise par le processus *AP ( )*, sous forme d'un message *diameter_response_finish*. Par la suite, une fois que le processus *Srv_Diameter ( )* l'aura reçu, il transitera vers le sous-état *Succès1* et pourra finalement émettre le message *diameter_success* vers le processus *AP ( )* qui est synonyme de succès définitif de l'authentification et, par la même occasion, permettra de réinitialiser la machine à états finis du processus *Srv_Diameter( )* avec une transition vers l'état de départ, en l'occurrence *Initialisation*.

2) Si l'authentification échoue et plus particulièrement si le système a fait l'objet d'une attaque, le processus *Srv_Diameter ( )* envoie un message *diameter_alert* au processus *AP ( )*, qui contient la cause de l'échec de l'authentification pour qu'il en soit informé et puisse le transmettre à l'usager mobile. Par la suite, le processus *Srv_Diameter ( )* transite vers l'état *Echec*. Toutefois, la communication n'est pas encore interrompue et le processus *Srv_Diameter ( )* doit attendre un acquittement du message d'erreur qu'il a envoyé et qui consiste en la réception d'un message *diameter_response_alert* envoyé par le processus *AP ( )*. Une fois ce message reçu, le processus

Srv_Diameter ( ) transite vers le sous-état *Echec1* depuis lequel il pourra envoyer au processus *AP ( )* le message d'échec final nommé *diameter_failure*. Ce dernier, mettra fin à la communication et réinitialisera la machine à états finis du processus *Srv_Diameter ( )* avec une transition vers l'état de départ qui n'est autre que *Initialisation*.

Nous présentons maintenant les modèles que nous avons réalisés pour simuler le comportement de notre mécanisme en cas d'attaques relatives à l'authentification, en l'occurrence une attaque de type usurpation d'identité ou usurpation d'authentification, et une attaque de type « man in the middle » ou mascarade. Nous avons modélisé un usager mobile malhonnête et un point d'accès malicieux.

Le point d'accès malicieux permet de simuler une attaque de type « man in the middle » ou mascarade. En effet, le point d'accès malicieux intervient entre l'usager mobile et le serveur d'authentification et essaie de se faire passer pour un point d'accès légitime afin d'intercepter des informations secrètes ou d'effectuer des opérations interdites ou encore d'avoir accès à des ressources protégées. Nous l'avons modélisé sous forme du processus *PAM ( )* pour Point d'Accès Malicieux. La Figure 4.4 présente ce modèle. Nous verrons à l'étape de validation comment notre mécanisme permet d'éviter cette attaque.

L'usager mobile malhonnête tente de s'authentifier en se faisant passer pour une autre personne. C'est une attaque de type usurpation d'identité ou usurpation d'authentification. Nous avons appelé ce processus *UM ( )* pour Usager Malhonnête. Son modèle est présenté à la Figure 4.5. L'usager malhonnête essaye d'usurper l'identité d'un usager mobile honnête afin d'avoir accès aux ressources protégés. Nous verrons à l'étape de validation comment notre mécanisme permet d'éviter cette attaque.

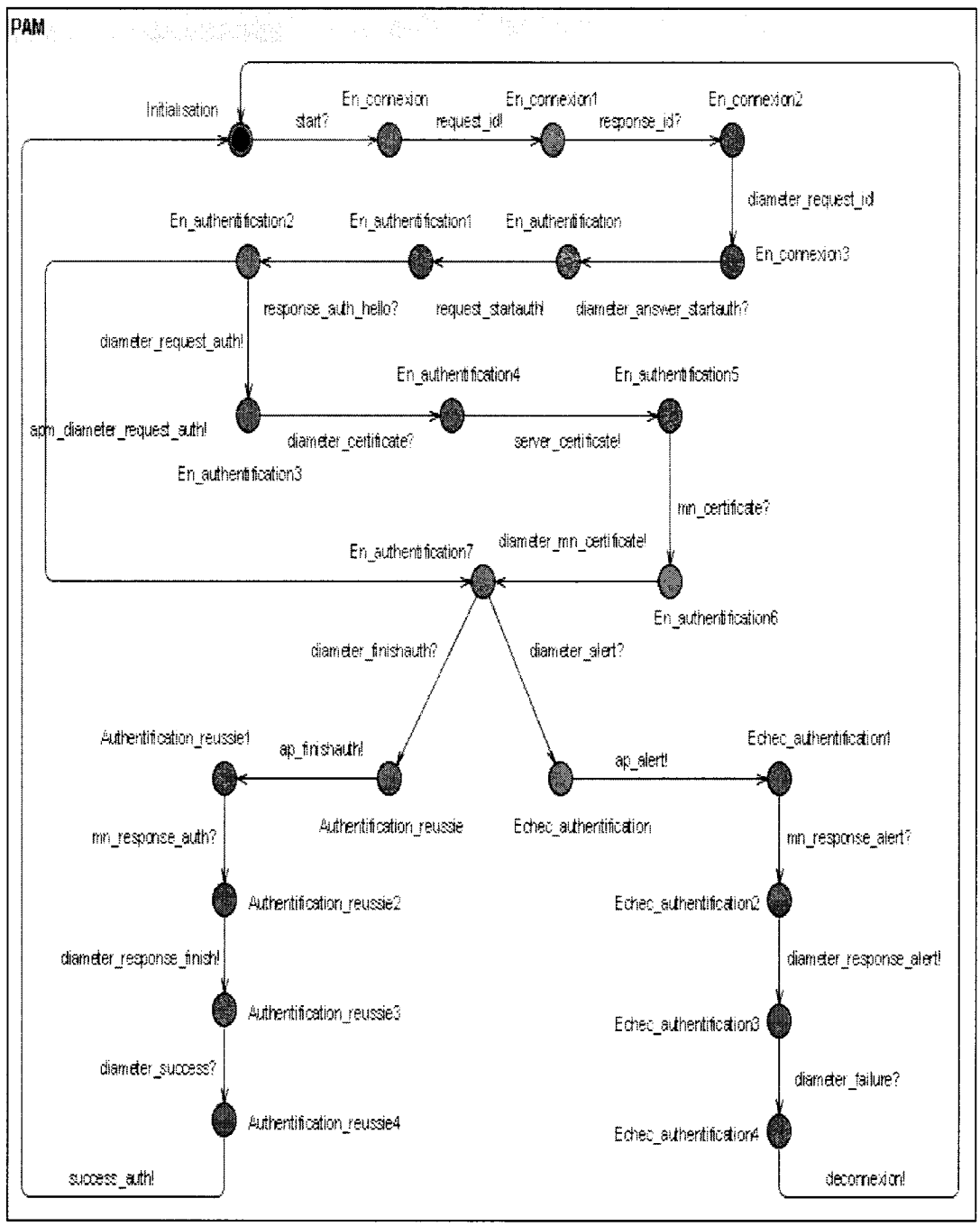

Figure 4.4 Modèle du processus PAM

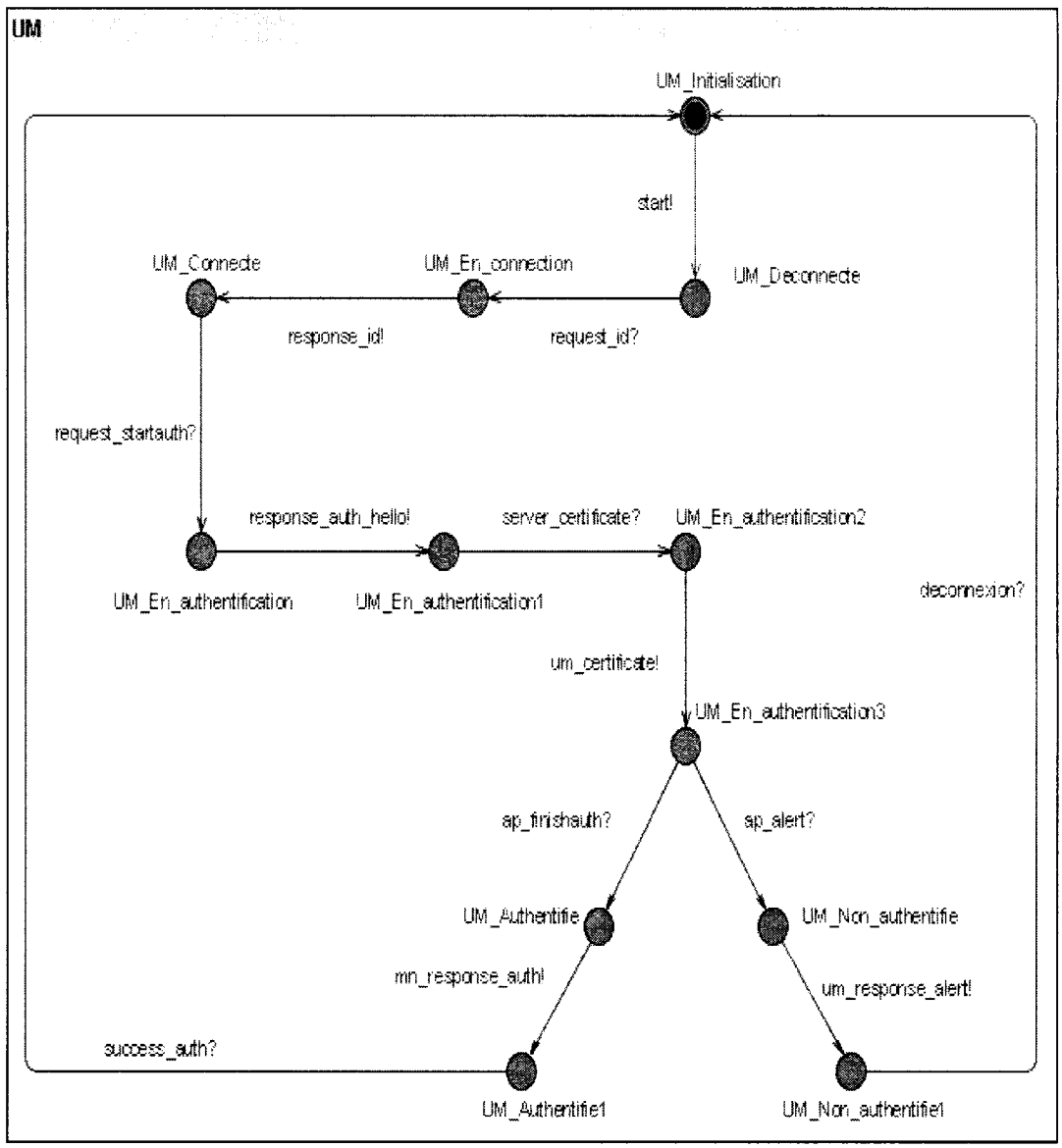

Figure 4.5 Modèle du processus UM

4.3 Validation du mécanisme

L'étape de validation consiste à programmer le modèle conçu à l'aide du « model-cheker » UPPAAL et à simuler son fonctionnement afin de s'assurer de sa conformité aux requis du mécanisme exposé au chapitre III. Cette étape est très importante, voire même cruciale car elle nous permettra de passer à la phase de

vérification des propriétés tout en étant intimement convaincu que la modélisation réalisée dans ce chapitre est conforme à la conception réalisée dans le chapitre précédent.

UPPAAL est muni d'une option d'exécution aléatoire qui ne fait pas intervenir l'utilisateur, ce qui est meilleur pour l'exactitude des résultats. Il génère aussi une trace d'exécution pour que l'utilisateur puisse suivre étape par étape l'évolution du système. Nous présentons par la suite les résultats de différentes exécutions des modèles que nous avons réalisés. La Figure 4.6 représente un aperçu de la trace d'exécution de la première phase du mécanisme.

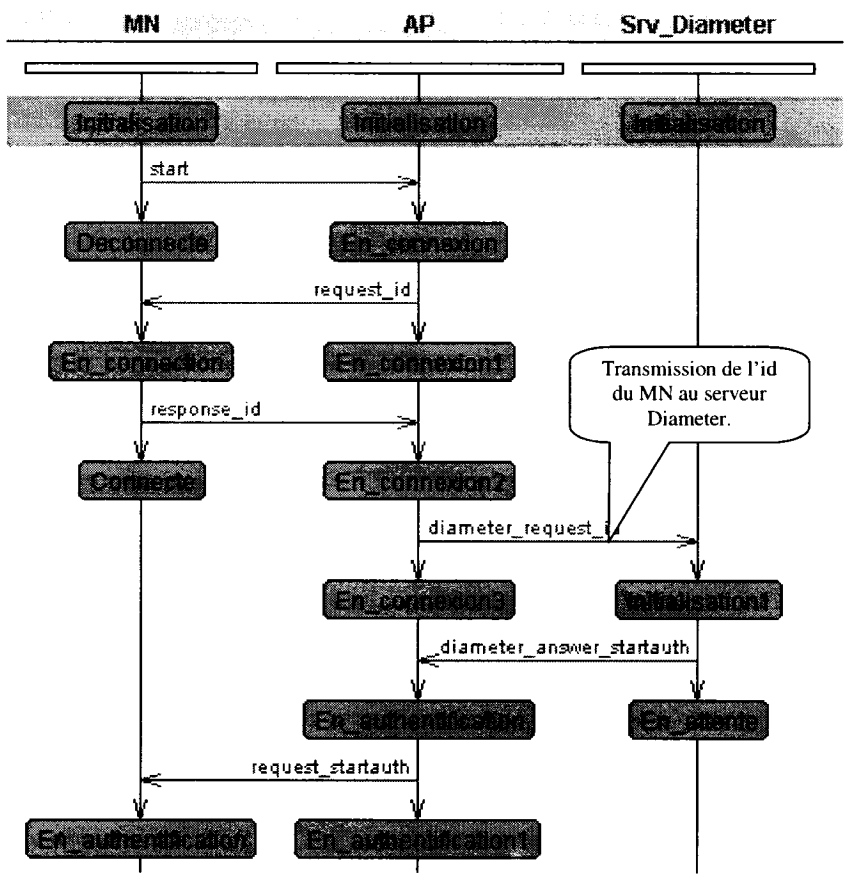

Figure 4.6 Trace d'exécution de la première phase du mécanisme

Nous voyons simultanément, pour chaque processus, les différents états par lesquels il passe et les messages échangés. L'usager mobile MN initie la communication avec le point d'accès AP. Ensuite, les négociations sur l'identité du MN ont lieu et son identifiant (id) est transmis au serveur d'authentification *Diameter*. Enfin, le serveur initie une session d'authentification en répondant à l'usager mobile via le point d'accès.

La Figure 4.7 représente la phase d'échange de certificats entre le MN et le serveur d'authentification *Diameter*. Le MN commence par demander au serveur de s'authentifier. Une fois qu'il reçoit son certificat, il envoie le sien.

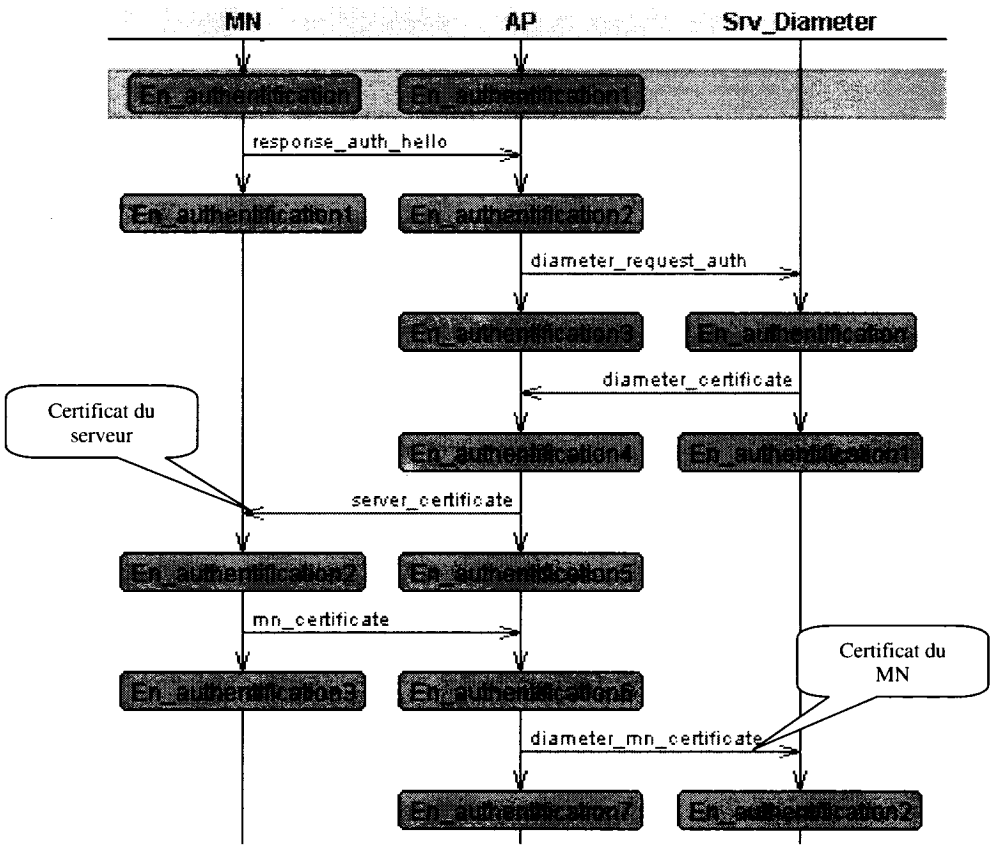

Figure 4.7 Trace d'exécution de la phase d'échange de certificats

Si l'usager mobile est honnête, le processus évolue normalement et tout se passe bien, l'authentification est donc réussie. La Figure 4.8 représente une trace d'exécution dans le cas de succès de l'authentification. L'usager mobile MN est correctement authentifié et il pourra avoir accès aux ressources protégées.

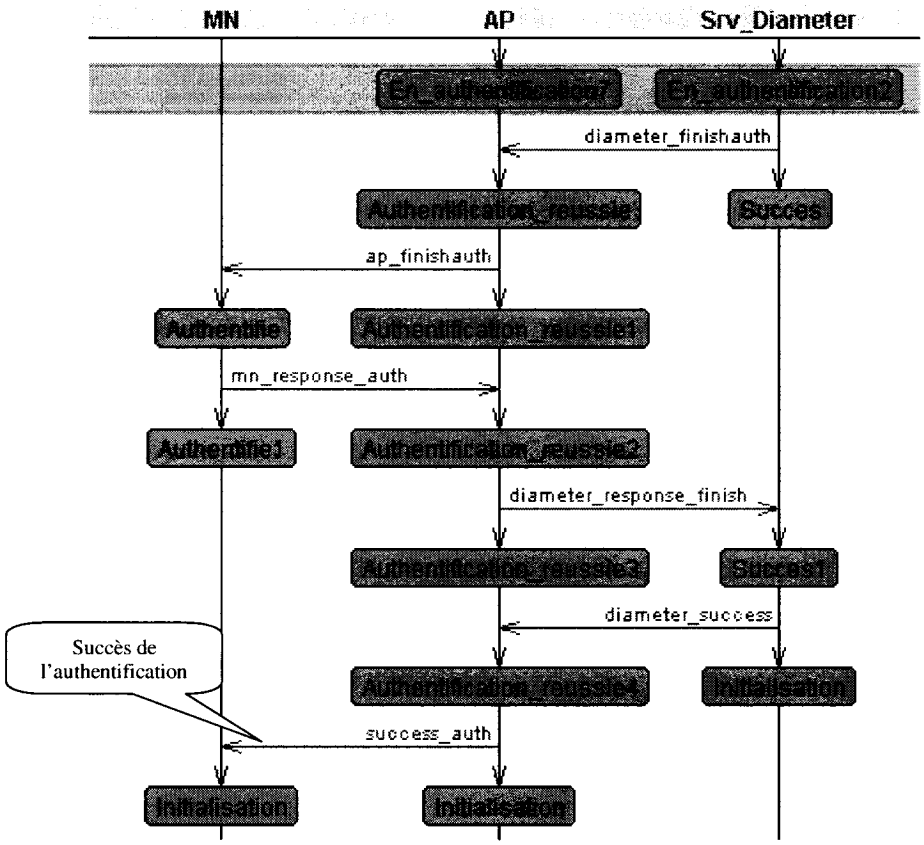

Figure 4.8 Trace d'exécution du succès de l'authentification

La Figure 4.9 représente la trace d'exécution d'une attaque de type usurpation d'authentification. Nous avons utilisé le modèle de l'usager malhonnête à travers le processus *UM ( )* afin de simuler cette attaque. Nous voyons que l'usager malhonnête fournit un certificat erroné, ce qui est détecté par le serveur *Diameter* et implique directement l'échec de l'authentification. Par la suite, les différents messages d'alerte sont échangés pour finalement aboutir à la déconnexion de l'usager malhonnête.

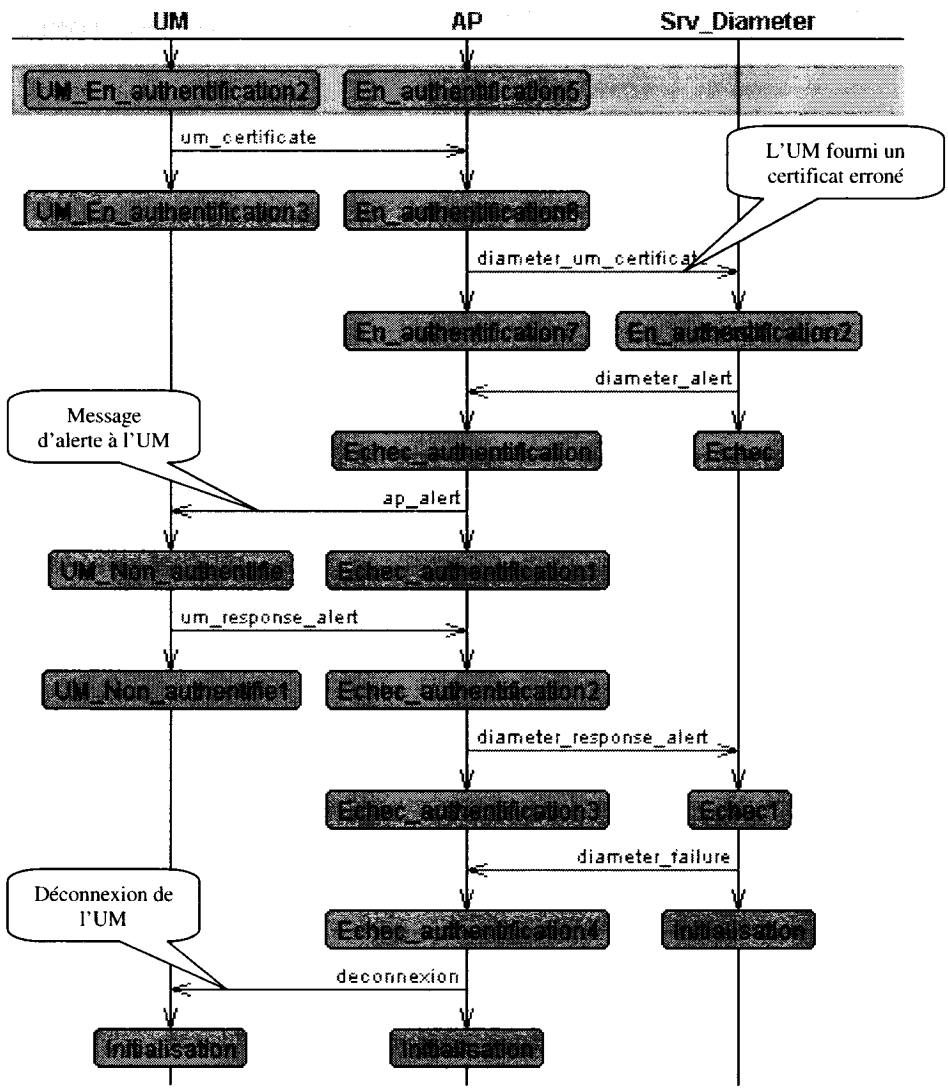

Figure 4.9 Trace d'exécution de l'attaque usurpation d'authentification

Nous avons par la suite procédé à la simulation de l'attaque de type mascarade ou « man in the middle ». Nous avons utilisé le modèle du point d'accès malicieux représenté par le processus *PAM ( )*. Nous voyons à la Figure 4.10 qu'au début le point d'accès malicieux se comporte comme un point d'accès légitime mais, lorsque le serveur *Diameter* reçoit le message de début d'échange de certificat, il détecte que c'est une

attaque et interrompt directement l'enchaînement normal du processus avec un message d'alerte, qui engendre par la suite la déconnexion de l'usager mobile.

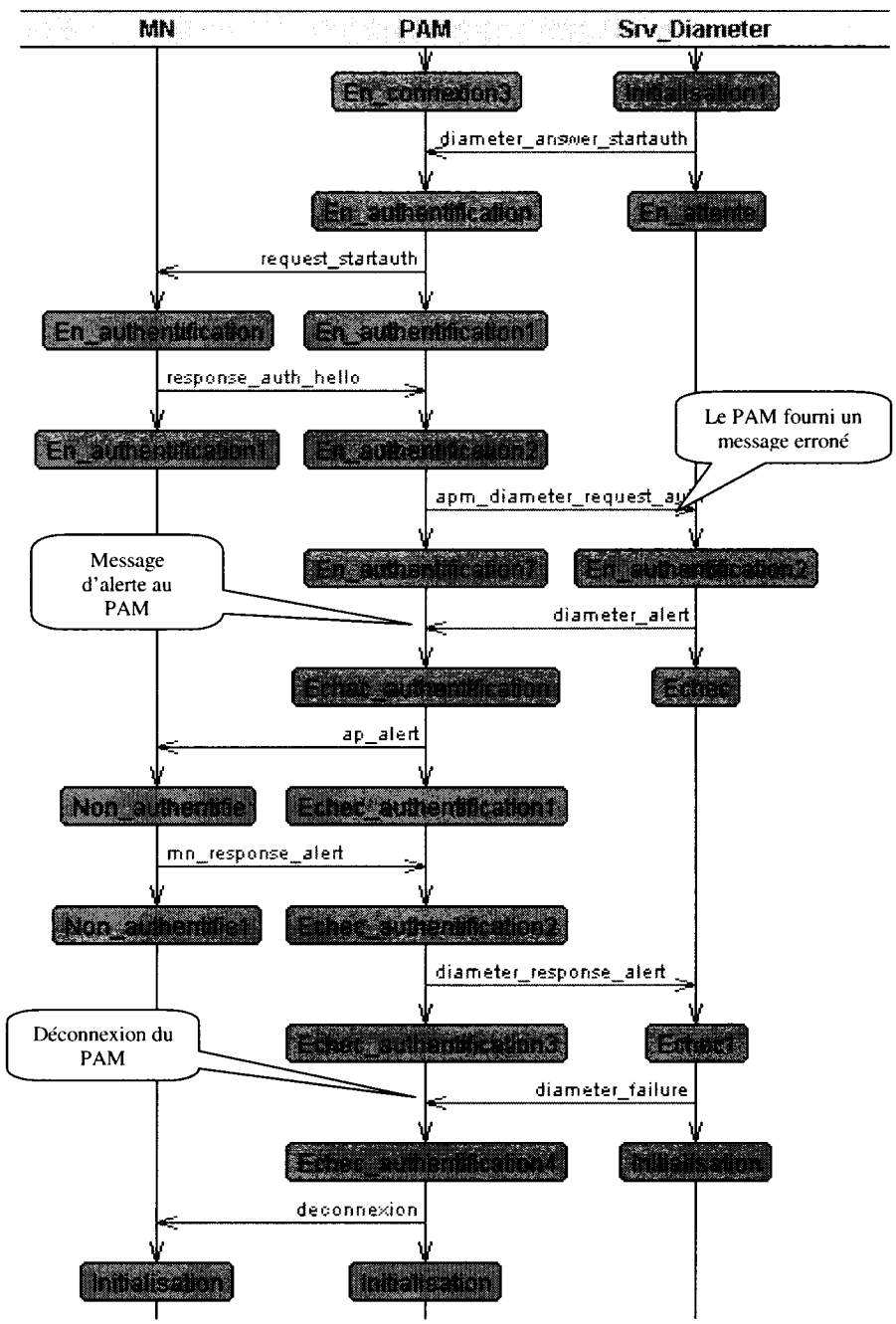

Figure 4.10 Trace d'exécution de l'attaque de type mascarade

En résumé, nous constatons que les résultats des différentes exécutions de nos modèles sont très satisfaisants aussi bien dans le cas d'attaques avec des intervenants malhonnêtes, que dans le cas sans attaques avec des intervenants honnêtes. L'interaction entre les différents processus modélisés est conforme à nos attentes. L'évolution globale du système dans ses différentes phases concorde avec notre modèle conceptuel présenté au chapitre précédent. Nous avons pu aussi vérifier que notre mécanisme permet d'éviter les attaques relatives à l'authentification et notamment l'attaque de type usurpation d'authentification et l'attaque de type mascarade ou « man in the middle ».

4.4 Vérification des propriétés du mécanisme

Suite à la modélisation de notre mécanisme et à son exécution dans différents cas de figure, notamment dans le cas d'attaques, nous entamons maintenant l'étape de vérification des propriétés. Cette étape intègre deux grandes tâches : la spécification des propriétés que notre système doit satisfaire et leur vérification.

La spécification des propriétés consiste à exprimer les contraintes auxquelles doit répondre le système à l'aide d'une logique. La vérification, quant à elle, consiste à exécuter tous les cas possibles du modèle pour vérifier si les propriétés spécifiées sont satisfaites. En d'autres termes, cela consiste à s'assurer que tous les états vérifiant les propriétés sont bien atteints en effectuant une recherche exhaustive qui couvre tous les comportements possibles du modèle.

Le premier outil dont nous avons besoin est une logique que nous utiliserons pour spécifier les propriétés à vérifier. En général, les logiques utilisent une combinaison de propositions par des connecteurs logiques tels que la négation ou la conjonction, afin d'exprimer des propriétés qui qualifient l'enchaînement des états (exécutions ou évolution du système au cours du temps à partir d'un état particulier). Plusieurs types de logiques existent dans la littérature et peuvent être utilisées. Citons par exemple les logiques PLTL, CTL et TCTL. Ces logiques se distinguent par l'ensemble des opérateurs temporels qu'ils utilisent et des objets sur lesquels ils opèrent.

La logique qui a été utilisée pour notre vérification est la CTL (Computation Tree Logique). Les opérateurs utilisés pour cette logique sont les suivants :

- Opérateur G ([]) permet d'exprimer que tous les états futurs, y compris l'état courant, vérifient une propriété ;
- Opérateur X (O) permet de spécifier qu'une propriété sera vérifiée à l'état qui suit l'état considéré ;
- Opérateur F (<>) permet d'indiquer qu'un état vérifiera fatalement une propriété dans le futur ;
- Opérateur U permet d'énoncer qu'une propriété sera vérifiée tant qu'une autre ne l'est pas ;
- Opérateur W permet d'indiquer qu'une propriété sera toujours vérifiée à moins qu'une autre ne le soit.

Outre les opérateurs temporels, les logiques temporelles disposent de quantificateurs de chemins, désignés par E et A, pour exprimer l'aspect arborescent des propriétés :

- Quantificateur A permet d'énoncer qu'une propriété est vérifiée par toutes les séquences d'états de l'arbre d'exécution débutant à l'état courant ;
- Quantificateur E permet de spécifier qu'une propriété est vérifiée par au moins une séquence partant de l'état courant de l'arbre.

Les propriétés à vérifier sont généralement classées en quatre grandes catégories qui sont : absence de blocage, accessibilité, sûreté et vivacité. Nous les présentons à la suite avec pour chaque catégorie les propriétés relatives à notre système que nous avons vérifiées.

Absence de blocage

Cela veut dire que le système ne peut pas se trouver dans une situation où il lui est impossible d'évoluer. Elle s'exprime en logique *CTL* à l'aide de l'opérateur **[]** et du quantificateur **A**. Avec UPPAAL, on utilise le mot réservé **not deadlock** comme suit :

A[ ] not deadlock → La propriété est satisfaite pour notre système.

Propriétés d'accessibilité

Ce sont les propriétés qui pourraient s'énoncer de la manière suivante : existe-t-il un chemin commençant à l'état initial du système et tel que la propriété φ est éventuellement satisfaite. Elles s'expriment en CTL sous la forme **E** ◇φ et avec UPPAAL sous la forme **E** <> φ. Nous pouvons citer comme exemple :

« **Est ce qu'il est possible que l'usager mobile MN soit authentifié ?** »

qui se traduit en CTL : **E** ◇**MN.Authentifie**

et avec UPPAAL : **E<> MN.Authentifie**

→ La propriété est satisfaite pour notre système.

« **Est ce qu'il est possible que l'usager mobile MN ne soit pas authentifié ?** »

qui se traduit en CTL : **E** ◇**MN. Non_authentifie**

et avec UPPAAL : **E<> MN.Non_authentifie**

→ La propriété est satisfaite pour notre système.

« **Est ce qu'il est possible que l'usager mobile MN soit simultanément authentifié et non authentifié ?** »

qui se traduit en CTL : **E ◇MN.Authentifie and MN.Non_authentifie**

et avec UPPAAL : **E<> MN.Authentifie && MN.Non_authentifie**

→ La propriété n'est pas satisfaite pour notre système.

« **Est ce qu'il est possible qu'un usager mobile malhonnête soit authentifié** »

qui se traduit en CTL : **E ◇UM.UM_Authentifie**

et avec UPPAAL : **E<> UM.UM_Authentifie**

→ La propriété n'est pas satisfaite pour notre système.

« **Est ce qu'il est possible qu'un usager mobile soit authentifié lorsqu'il y a un point d'accès malicieux** »

qui se traduit en CTL : **E ◇apm==1 and PAM.Authentification_reussie**

et avec UPPAAL : **E<> apm==1 && PAM.Authentification_reussie**

→ La propriété n'est pas satisfaite pour notre système.

Ce type de propriété est très intéressant aussi puisqu'il nous permet de vérifier si un intervenant peut envoyer un message. Par exemple :

« **Est ce qu'il est possible que le serveur *Diameter* puisse envoyer le message de début d'authentification ?** »

qui se traduit en CTL : **E ◇Srv_Diameter.En_attente**

et avec UPPAAL : **E<> Srv_Diameter.En_attente**

→ La propriété est satisfaite pour notre système.

Il est aussi possible de vérifier si un message a bien été reçu par un intervenant du mécanisme, par exemple :

« **Est ce qu'il est possible que le point d'accès AP puisse recevoir le message de succès de l'authentification ?** »

qui se traduit en CTL : **E ◇AP.Authentification_reussie**

et avec UPPAAL : **E<> AP.Authentification_reussie**

→ La propriété est satisfaite pour notre système.

Tableau 4.1 Synthèse des propriétés d'accessibilité vérifiées

Propriété	Satisfaite	Non Satisfaite
1) L'usager mobile est authentifié	X	
2) L'usager mobile n'est pas authentifié	X	
3) L'usager mobile est simultanément authentifié et non authentifié		X
4) Un usager mobile malhonnête est authentifié		X
5) Un usager mobile est authentifié lorsqu'il y a un point d'accès malicieux.		X
6) Le serveur *Diameter* peut envoyer le message de début d'authentification.	X	
7) le point d'accès reçoit le message de succès de l'authentification	X	

Propriétés de sûreté

Ce type de propriété peut avoir deux formulations distinctes. Par exemple, si une propriété φ devrait être vérifiée pour tous les états atteignables et ce, quel que soit l'arbre d'exécution, elle est symbolisé en CTL par **A** □ φ, et avec UPPAAL par **A [ ]** φ. Par contre, s'il suffit d'avoir au moins un arbre d'exécution pour lequel φ est toujours vrai, alors on utilise **E** □ φ en CTL et **E[ ]** φ avec UPPAAL. Par exemple, voici quelques propriétés :

« **Est ce qu'il est toujours vrai qu'un usager mobile malhonnête ne pourra pas être authentifié ?** »

qui se traduit en CTL : **A** □ ¬ **UM.UM_Authentifie**

et avec UPPAAL : **A[ ] not UM.UM_Authentifie**

→ La propriété est satisfaite pour notre système.

« **Est ce qu'il est toujours vrai que si l'authentification échoue au niveau du serveur *Diameter* alors l'usager mobile ne pourra pas être authentifié ?** »

qui se traduit en CTL : **A** □ **Srv_Diameter.Echec imply** ¬ **MN.Authentifie**

et avec UPPAAL : **A[ ] Srv_Diameter.Echec imply not MN.Authentifie**

→ La propriété est satisfaite pour notre système.

« **Est ce qu'il est toujours vrai que si l'authentification réussit au niveau du serveur *Diameter*, alors l'usager mobile MN ne pourra pas être dans l'état non authentifié ?** »

qui se traduit en CTL : **A** □ **Srv_Diameter.succes imply** ¬ **MN.Non_authentifie**

et avec UPPAAL : **A[ ] Srv_Diameter.succes imply not MN.Non_authentifie**

→ La propriété est satisfaite pour notre système.

Tableau 4.2 Synthèse des propriétés de sûreté vérifiées

Propriété	Satisfaite	Non Satisfaite
1) Un usager mobile malhonnête n'est jamais authentifié	X	
2) Si l'authentification échoue au niveau du serveur *Diameter* alors l'usager mobile ne pourra pas être authentifié	X	
3) Si l'authentification réussit au niveau du serveur *Diameter* alors l'usager mobile ne pourra pas être dans l'état non authentifié	X	

Propriétés de vivacité

Ce sont des propriétés qui énoncent que, si un ensemble de conditions φ sont réunies, alors la propriété ψ finira par avoir lieu. Elles sont symbolisées en CTL par $\varphi \rightarrow \psi$ et avec UPPAAL par φ --> ψ. Voici quelques exemples de ce type de propriétés :

« **Est ce qu'il est toujours vrai que si le serveur *Diameter* décide que l'authentification a échoué, alors l'usager mobile MN sera obligatoirement déconnecté ?** »

qui se traduit en CTL : **Srv_Diameter.Echec1 → MN.Non_authentifie1**

et avec UPPAAL : **Srv_Diameter.Echec1 --> MN.Non_authentifie1**

→ La propriété est satisfaite pour notre système.

« **Est ce qu'il est toujours vrai que si le serveur *Diameter* décide que l'authentification a réussi, alors l'usager mobile MN sera obligatoirement authentifié ?** »

qui se traduit en CTL : **Srv_Diameter.Succes1 → MN.Authentifie1**

et avec UPPAAL : **Srv_Diameter.Succes1 --> MN.Authentifie1**

→ La propriété est satisfaite pour notre système.

« **Est ce qu'il est toujours vrai qu'une attaque d'un point d'accès malicieux est détectée ?** »

qui se traduit en CTL : apm==1 and PAM.Echec_authentification → Srv_Diameter.Echec

et avec UPPAAL : **apm==1 && PAM.Echec_authentification --> Srv_Diameter.Echec**

→ La propriété est satisfaite pour notre système.

Ce type de propriété nous permet aussi de vérifier que chaque message qui a été envoyé doit être reçu. Par exemple :

« **Est ce qu'il est toujours vrai que si l'usager mobile MN a reçu le message ap_alert, alors le point d'accès AP a envoyé ce message ?** »

qui se traduit en CTL: **MN.Non_authentifié → AP.Echec_authentification1**

et avec UPPAAL : **MN.Non_authentifié --> AP.Echec_authentification1**

→ La propriété est satisfaite pour notre système.

Tableau 4.3 Synthèse des propriétés de vivacité vérifiées

Propriété	Satisfaite	Non Satisfaite
1) Si le serveur *Diameter* décide que l'authentification a échoué alors l'usager mobile sera obligatoirement déconnecté	X	
2) Si le serveur *Diameter* décide que l'authentification a réussie, alors l'usager mobile sera obligatoirement authentifié	X	
3) Une attaque d'un point d'accès malicieux est toujours détectée	X	
4) Si l'usager mobile a reçu le message ap_alert, alors le point d'accès a envoyé ce message	X	

L'analyse des résultats de la validation de notre mécanisme révèle que nos requis définis au niveau conceptuel sont bien atteins. En effet, l'exécution des différents modèles que nous avons réalisés décrit exactement le comportement de notre mécanisme comme spécifié à l'étape conceptuelle. L'authentification ne s'effectue plus au niveau du point d'accès mais au niveau du serveur d'authentification *Diameter*. Cela permet d'offrir une meilleure mobilité puisque l'usager n'est plus limité à se connecter via une sélection restrictive de points d'accès qui sont correctement configurés pour l'authentifier. Cela dit, en cas d'échec de l'authentification, au lieu de couper directement la communication, le serveur informe l'usager mobile de la cause de l'échec afin d'éviter que le même problème se répète de nouveau. De plus, la propriété d'absence de blocage que nous avons vérifiée confirme que, quelle que soit l'exécution

du système, l'interaction entre les différents processus modélisés n'engendre aucun blocage.

Ensuite, nous avons vu qu'en cas d'attaque, l'évolution globale du système dans ces différentes phases n'est aucunement altérée. Par exemple, la propriété 4 d'accessibilité, les propriétés 1 et 2 de sûreté et la propriété 3 de vivacité que nous avons vérifiées prouvent que notre mécanisme permet de détecter et d'éviter ces attaques. Plus spécifiquement, l'attaque d'usurpation d'identité et l'attaque du point d'accès malicieux sont détectées grâce à l'utilisation des notions de certificats numériques et d'authentification mutuelle au sein de notre mécanisme. Enfin, la propriété 3 d'accessibilité, la propriété 1 de vivacité et la propriété 3 de sûreté prouvent que, quelle que soit l'exécution du système, aucune situation d'incohérence entre les différents états n'est engendrée.

CHAPITRE V

CONCLUSION

Dans ce dernier chapitre, nous présentons, tout d'abord, une synthèse générale de nos travaux de recherche. Ensuite, nous abordons les limitations de nos travaux. Enfin, nous terminons par présenter quelques indications vers des orientations de recherche futures.

5.1 Synthèse des travaux

Dans nos travaux, nous nous sommes intéressés à l'une des catégories de réseaux qui a connu l'un des plus grands développements ces dernières années : les réseaux locaux sans fil. La raison pour laquelle ce type de réseau a connu une si grande expansion est qu'il offre de nombreux avantages comme par exemple la facilité de déploiement, une meilleure mobilité des usagers ou encore des coûts de mise en place assez faibles. Par contre, l'inconvénient majeur qu'il introduit est la sécurité des communications. Comme nous l'avons bien illustré dans notre revue de littérature, les problèmes de sécurité dans les réseaux locaux sans fil sont nombreux, diversifiés et complexes. Cependant, ils sont généralement classifiés en deux grandes catégories : d'une part, les problèmes d'ordre cryptographique regroupant le chiffrement des données, la gestion des clés d'encryption et l'authentification et d'autre part, les problèmes d'ordre architectural. Prétendre résoudre tous ces problèmes serait utopique. C'est dans ce contexte que nous nous sommes concentrés sur un aspect de ces problèmes, et non le moindre, qui est l'authentification.

L'authentification dans les réseaux locaux sans fil a, pendant une longue période, été basée sur WEP et très récemment sur le nouveau standard 802.11i. C'est pour cette raison que nous avons, tout d'abord, commencé par analyser conceptuellement et techniquement les modèles et protocoles présentés dans WEP et dans le nouveau

standard 802.11i, ce qui nous a permis de caractériser leurs principales failles de sécurité et de mettre en évidence les nouveaux défis qu'ils soulèvent.

À la lumière de ces résultats et afin de pallier ces lacunes, nous avons proposé un mécanisme permettant d'assurer une authentification sécurisée des usagers mobiles se déplaçant dans des réseaux locaux sans fil. Le mécanisme que nous avons proposé est constitué essentiellement de trois grandes phases : la première phase consiste en la découverte par un usager mobile du réseau et des paramètres de sécurité offerts par un point d'accès. La seconde phase consiste en une association 802.11 entre cet usager mobile et le point d'accès considéré avec négociation des paramètres de sécurité. La troisième et dernière phase consiste en une authentification *Diameter* de bout en bout. Par ailleurs, les principes du mécanisme que nous avons spécifié et conçu constituent en soi une contribution originale. En effet, nous avons proposé une architecture trois tiers avec un serveur d'authentification à part entière, ce qui nous a différencié des architectures actuelles de type deux tiers. Nous avons introduit un concept innovateur d'authentification mutuelle entre les usagers mobiles et le serveur d'authentification. Nous avons proposé un concept nouveau de port contrôlé qui permet de bloquer le trafic en cas d'échec du processus d'authentification. Afin d'assurer un degré de confidentialité accru, nous avons utilisé une notion cryptographique de pointe, en l'occurrence les certificats numériques. De plus, notre mécanisme est le premier à avoir utilisé un serveur *Diameter* dans le contexte des réseaux locaux sans fil. Par ailleurs, en cas d'échec du processus d'authentification notre mécanisme permet d'informer l'usager mobile de la cause de l'échec afin d'éviter qu'elle se répète de nouveau ultérieurement.

L'étape suivante était de valider formellement notre mécanisme afin de donner encore plus de valeur à notre travail. Pour ce faire, nous avons, tout d'abord, modélisé à l'aide d'une machine à états finis chaque intervenant dans le processus d'authentification de notre mécanisme, c'est-à-dire l'usager mobile, le point d'accès et le serveur d'authentification en essayant de reproduire scrupuleusement le comportement de notre modèle conceptuel. Nous avons aussi modélisé un usager mobile malhonnête et un point d'accès malicieux afin de simuler le comportement de notre mécanisme en cas

d'attaques. Les résultats ont été très satisfaisants. En effet, les traces d'exécution générées par le *model-cheker* UPPAAL et l'évolution globale du système dans ses différentes phases se sont révélées conformes à notre modèle conceptuel. De plus, les différents cas d'attaque étaient automatiquement détectés et évités. Enfin, à l'aide de CTL, une logique formelle, nous avons exprimé un ensemble de propriétés d'accessibilité, de vivacité, de sûreté et de non blocage représentant des contraintes auxquels doit répondre notre système quelle que soit son exécution. La vérification effective de ces propriétés avec le *model-cheker* UPPAAL nous a permis de nous assurer indéniablement de l'absence de blocage dans notre mécanisme, qu'aucune situation d'incohérence entre les différents états n'est engendrée et que toutes les attaques sont bien détectées et évitées, et ce, quelle que soit l'exécution possible et imaginable des modèles conçus.

5.2 Limitations des travaux

En dépit des résultats très satisfaisants de nos travaux, ils comportent certaines limitations. Tout d'abord, comme nous l'avons expliqué au chapitre trois, notre mécanisme est destiné à des réseaux locaux sans fil fonctionnant en mode infrastructure. Dans le cas de réseaux locaux sans fil fonctionnant en mode ad hoc, étant donné les caractéristiques intrinsèques à ce type de réseau, comme l'absence d'infrastructure fixe de communication impliquerait des adaptations à notre mécanisme.

Un autre point à approfondir serait le cas de relève horizontale. Dans nos travaux, notre mécanisme réinitialise une session d'authentification avec le nouveau point d'accès en cas de relève horizontale, que se soit une relève dans le même sous-réseau avec la même plage d'adresses IP ou bien que se soit une relève entre deux sous-réseaux différents avec des plages d'adresses IP distinctes, ce qui pourrait engendrer des délais de signalisation supplémentaires.

Finalement, étant donné que nous nous sommes concentrés sur les réseaux locaux sans fil, notre mécanisme ne prend pas en compte les cas de relève verticale, c'est

à dire une relève entre réseaux d'accès faisant intervenir des technologies différentes comme UMTS ou Bluetooth.

5.3 Orientations de recherches futures

La sécurité dans les réseaux locaux sans fil demeure un problème ouvert. Les pistes d'améliorations sont nombreuses. Des pistes de recherches futures pourraient découler directement des limitations que nous avons identifiées précédemment. Notamment, dans les réseaux locaux sans fil évoluant en mode ad hoc, l'absence d'infrastructure de communication entre les intervenants rend les problèmes de sécurité très diversifiés et très importants. Le cas de relève horizontale avec changement de sous-réseau et par conséquent d'adresse IP serait intéressant à étudier aussi. Ce genre de situation ferait intervenir simultanément une combinaison de protocoles de sécurité et de protocoles de gestion de mobilité. Dans un contexte plus général, le cas de relève vertical serait pertinent à étudier aussi. Vu la complexité croissante des architectures de réseaux de nouvelles générations, des cas de relève entre différentes technologies d'accès sont très envisageables. L'aspect de sécurité étant primordial, il engendre de nombreux défi de recherche. Finalement, on pourrait aussi envisager une mise en place d'une plate-forme de test ou encore de réaliser une émulation de réseaux locaux sans fil afin entre autres de vérifier la robustesse en cas d'attaques réelles ou encore d'évaluer le délai généré par la procédure d'authentification et d'étudier son impact sur de futures applications de voix sur IP ou de multimédia.

BIBLIOGRAPHIE

[1] IEEE Std 802.11-1997, *Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications.*

[2] PARK Joon S., DICOI Derrick. 2003. «WLAN Security: Current and Future». *IEEE Internet Computing,* vol. 7, no. 5, pp. 60-65.

[3] BORISOV Nikita, GOLDBERG Ian, WAGNER David. 2001. «Security of the WEP algorithm». [En ligne] http://www.isaac.cs.berkeley.edu/isaac/wep-faq.html (page consultée le 15 octobre 2004).

[4] BORISOV Nikita, GOLDBERG Ian, WAGNER David. 2001. «Intercepting Mobile Communications: The Insecurity of 802.11». *ACM. Seventh Annual International Conference on Mobile Computing And Networking: July 16-21, 2001, Roma, Italy, pp. 180-189.*

[5] RAMACHANDRAN Jay. 2002. *Designing Security Architecture Solutions.* [En ligne]. John Wiley & Sons (416 pages). http://library.books24x7.com/ (page consultée le 12 octobre 2004).

[6] CHECK POINT SOFTWARE TECHNOLOGIES. 2003. *Principles of Network Security.* [En ligne]. Check Point Software Technologies (382 pages). http://library.books24x7.com/ (page consultée le 17 octobre 2004).

[7] ARBAUGH William A, SHANKAR Narendar, WAN Justin Y.C. 2002. «Your 802.11 Wireless Network has No Clothes». *IEEE Wireless Communications Magazine*, vol. 9, no. 6, pp. 44-51.

[8] PETRONI Nick. L. Jr., ARBAUGH William A. 2003. «The Dangers of Mitigating Security Design Flaws : A Wireless Case Study ». *IEEE Security and Privacy*, vol. 1, no. 1, pp. 28-36.

[9] WILLIAMS Joseph. 2002. «Providing for Wireless LAN Security, Part 2». *IEEE IT Professional*, vol. 4, no. 6, pp. 44-48.

[10] WILLIAMS Joseph. 2001. «The IEEE 802.11b Security Problem, Part 1». *IEEE IT Professional*, vol. 3, no. 6, pp. 91-96.

[11] The White House. 2003. *The National Strategy to Secure Cyberspace.* [En ligne] http://www.whitehouse.gov/pcipb/ (page consultée le 15 décembre 2004).

[12] Virtual Private Network Consortium. [En ligne] http://www.vpnc.org/ (page consultée le 10 décembre 2004).

[13] Wi-Fi Alliance [En ligne] http://www.wi-fi.org/ (page consultée le 20 décembre 2004).

[14] IEEE Std 802.11i-2004, *Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications. Amendment 6: Medium Access Control (MAC) Security Enhancements.*

[15] MISHRA Arunesh, ARBAUGH William A. 2002. *An Initial Security analysis of the IEEE 802.1x Standard.* Maryland: Department of Computer Science University of Maryland. 12 pages. CS-TR-4328 UMIACS-TR-2002-10.

[16] IEEE Std 802.1X-2001, *Port-Based Network Access Control for Local and Metropolitan Area Network.*

[17] CALHOUN Pat, LOUGHNEY John, ARKKO Jari, GUTTMAN Erik, ZORN Glen. 2003. *Diameter Base Protocol.* IETF RFC3588, 147 pages.

[18] RIGNEY Carl, RUBENS Allan, SIMPSON William, WILLENS Steve. 2000. *Remote Authentication Dial In User Service (RADIUS).* IETF RFC2865, 76 pages.

[19] ABOBA Bernard, BLUNK Larry, VOLLBRECHT John, CARLSON James, LEVKOWETZ Henrik. 2004. *Extensible Authentication Protocol (EAP).* IETF RFC3748, 67 pages.

[20] The InteropNet Labs (iLabs). 2004. *LAN Access Security Initiative* [En ligne] http://www.ilabs.interop.net/details?topic=LANSec (page consultée le 25 décembre 2004).

[21] SORMAN Matija, KOVAC Tomislav, MAUROVIC Damir. 2004. «Implementing Improved Wlan Security». *46th International Symposium Electronics in Marine (ELMAR 2004), Croatia, pp. 229-234.*

[22] ABOBA Bernard, CALHOUN Pat, GLASS Steven, HILLER Tom, MCCANN Peter. 2000. *Criteria for Evaluating AAA Protocols for Network Access.* IETF RFC2989, 28 pages.

[23] MITTON David, STJOHNS Michael, BARKLEY Stuart. 2001. *Authentication, Authorization, and Accounting Protocol Evaluation.* IETF RFC3127, 84 pages.

[24] ARUNESH Mishra, SHIN Min Ho, PETRONI Nick, CLANCY Charles, ARBAUGH William. 2004. «Proactive Key Distribution Using Neighbor Graphs». *IEEE Wireless Communications*, vol. 11, no. 1, pp. 26-36.

[25] STANLEY Dorothy, WALKER Jesse, ABOBA Bernard. 2005. *EAP Method Requirements for Wireless LAN.* IETF RFC4017, 11 pages.

[26] UPPAAL. 2005. [En ligne] www.uppaal.com (page consultée le 20 mars 2005).

[27] PERKINS Charles, PATIL Basavaraj, ROBERTS Phil. 2002. *IP Mobility Support for IPv4.* IETF RFC3344, 99 pages.

[28] ARBAUGH William. 2003. «Wireless Security Is Different». *IEEE Computer*, vol. 36, no. 8, pp. 99-101.

[29] POTTER Bruce. 2003. «Wireless Security's Future». *IEEE Security and Privacy*, vol. 1, no. 4, pp. 68-72.

[30] UPKAR Varshney. 2003. «The Status and Future of 802.11-based WLANs». *IEEE Computer*, vol. 36, no. 6, pp. 102-105.

[31] SONG Yubo, HU Aiqun. 2003. «The authentication in public WLAN when access controller deployed». *IEEE. International Conference on Neural Networks and Signal Processing: December 14-17, 2003, Nanjing, China, pp. 1666-1669.*